

---

# **XmlAutoGo Documentation**

***Release v0.0.0***

**[www.freeol.cn](http://www.freeol.cn)**

**Oct 25, 2018**



<b>1</b>	<b>What is XmlAutoGo?</b>	<b>1</b>
1.1	Features . . . . .	1
1.2	Command . . . . .	2
1.3	Download . . . . .	2
<b>2</b>	<b>Xml scripts tags</b>	<b>3</b>
2.1	Excel tags . . . . .	3
2.1.1	<XLS_EXCEL/> . . . . .	3
2.1.2	<XLS_IMG/> . . . . .	3
2.1.3	<XLS_SHEET/> . . . . .	4
2.1.4	<XLS_VALUE/> . . . . .	4
2.2	Lang tags . . . . .	4
2.2.1	<ADD/> . . . . .	4
2.2.2	<BREAK/> . . . . .	5
2.2.3	<CALL/> . . . . .	5
2.2.4	<CALL_EXEC/> . . . . .	5
2.2.5	<CONTINUE/> . . . . .	5
2.2.6	<COROUTINES/> . . . . .	6
2.2.7	<COROUTINE_FUNC/> . . . . .	6
2.2.8	<COROUTINE_EXEC/> . . . . .	6
2.2.9	<DEBUG/> . . . . .	6
2.2.10	<DOWNLOAD/> . . . . .	7
2.2.11	<ELIF/> . . . . .	7
2.2.12	<ELSE/> . . . . .	7
2.2.13	<EVAL/> . . . . .	7
2.2.14	<F2V/> . . . . .	7
2.2.15	<FOR/> . . . . .	8
2.2.16	<FUNC/> . . . . .	8
2.2.17	<IF/> . . . . .	9
2.2.18	<IMPORT/> . . . . .	9
2.2.19	<INSERT/> . . . . .	9
2.2.20	<MODULE/> . . . . .	9
2.2.21	<PUT/> . . . . .	10
2.2.22	<RETURN/> . . . . .	10
2.2.23	<SCRIPT/> . . . . .	11
2.2.24	<SET/> . . . . .	11

2.2.25	<SLEEP/>	11
2.2.26	<VARIABLE/>	11
2.2.27	<WHEN/>	12
2.2.28	<ZIP/>	12
2.3	UI tags	12
2.3.1	<UI_DLG/>	12
2.3.2	<UI_RUN_OVER/>	12
2.4	Webdriver tags	13
2.4.1	<D_ADD_COOKIE/>	13
2.4.2	<D_CLICKABLE/>	13
2.4.3	<D_DOWNLOAD/>	13
2.4.4	<D_INPUT/>	13
2.4.5	<D_JAVASCRIPT/>	14
2.4.6	<D_REFRESH/>	14
2.4.7	<D_SCREENSHOT/>	14
2.4.8	<D_SLEEP/>	14
2.4.9	<D_SWITCH_WIN/>	15
2.4.10	<D_TEXT_AS_VAR/>	15
2.4.11	<D_QUIT/>	15
2.4.12	<D_UPLOAD/>	15
2.4.13	<D_URL/>	16
2.4.14	<D_WAIT_FOR/>	16
2.5	Windows OS tags	16
2.5.1	<W_CURSOR_CLICK_LEFT/>	16
2.5.2	<W_CURSOR_CLICK_RIGHT/>	16
2.5.3	<W_CURSOR_POS/>	16
2.5.4	<W_DIRS_CLEAR/>	17
2.5.5	<W_DIRS_CLEAR_ALL/>	17
2.5.6	<W_FILE_COPY/>	17
2.5.7	<W_FILE_DEL/>	17
2.5.8	<W_FILE_MOVE/>	17
2.5.9	<W_FILE_READ/>	18
2.5.10	<W_FILE_RENAME/>	18
<b>3</b>	<b>Examples</b>	<b>19</b>
<b>4</b>	<b>Github</b>	<b>21</b>
<b>5</b>	<b>About us</b>	<b>23</b>
5.1	www.freeol.cn	23
5.2	Mail	23
<b>6</b>	<b>Updates</b>	<b>25</b>
6.1	version.0.0.14(Beta)	25
6.2	version.0.0.0(Release)	25

# CHAPTER 1

---

## What is XmlAutoGo?

---



XmlAutoGo is a tools that can be used to control browser(based on [Selenium 3.14.0](#)) and win32API on windows.

### 1.1 Features

- 1.Auto control chrome/ie/edge browser with the each webdriver.
- 2.Use xml to be scripts program files.
- 3.The xml script program support variables/method/module and so on.
- 4.It can use to upload files with browser.
- 5.Auto download files with browser.
- 6.There are two way to run the XmlAutoGo.**

- Click the XmlAutoGo.exe to use with windows UI.
- Run the command to run scripts program.

## 1.2 Command

```
$ XmlAutoGo.exe Chrome testScript.xml
```

## 1.3 Download

```
version0.0-beta.14
```

```
version0.0.0
```

>>Tags

## 2.1 Excel tags

Excel tags is the excel file's controller.

### 2.1.1 <XLS\_EXCEL/>

**summary** Defined excel's file(.xls).

eg.

```
<xls_excel name="test.xlsx" path="{!excel_path}">
  <xls_sheet index="0" title="test1">
    <xls_value cell="A1">test excel value</xls_value>
    <xls_img cell="A5" width="255" height="50">{!test1}</xls_img>
  </xls_sheet>
  <xls_sheet index="1">
    <xls_img cell="A1" width="255" height="50">{!test1}</xls_img>
    <xls_img cell="A55" width="255" height="50">{!test2}</xls_img>
  </xls_sheet>
</xls_excel>
```

#### notes

“name” is the excel's name.

“path” is the excel's path.

### 2.1.2 <XLS\_IMG/>

**summary** Insert a img into the excel.

eg.

```
<xls_img cell="A5" width="255" height="50">d:/test.png</xls_img>
```

#### notes

“cell” is the position of the sheet.

“width” is the width of the image.

“height” is the height of the image.

The tag’s content is the image’s local path.

### 2.1.3 <XLS\_SHEET/>

**summary** Defined a sheet in the excel file.

eg.

```
<xls_sheet index="0" title="test1">
  <xls_value cell="A1">test excel value</xls_value>
  <xls_img cell="A5" width="255" height="50">{!test1}</xls_img>
</xls_sheet>
```

#### notes

“index” is the index of the excel’s sheet.

“title” is the sheet’s name.

### 2.1.4 <XLS\_VALUE/>

**summary** Insert text into the excel.

eg.

```
<xls_value cell="A1">test excel value</xls_value>
```

#### notes

“cell” is the position of the sheet.

The tag’s content is the input text.

>>Tags

## 2.2 Lang tags

Lang tags is the xml scripts’ controller.

### 2.2.1 <ADD/>

**summary** Add item into the end of a list or array variable.

eg.

```
<add id="t2">"add is ok"</add>
```



**notes** “id” is the variable’s name

## 2.2.2 <BREAK/>

**summary** Breaks out of the innermost enclosing “<for/>”.

eg.

```
<for range="range(1,4)">
  <if>
    <when test="{!index}==2">
      <break/>
    </when>
  </if>
<d_javascript>console.log('{!index}')

```

**notes** none

## 2.2.3 <CALL/>

**summary** Call the method or module’s method

eg.

```
<call name="test_func" parameters="{ 'a':2, 'b':1 }"></call>
<call name="myModule.func1" parameters="{ 'a':2, 'b':1 }"></call>
```

**notes**

“name” is the method’s name, the method’s parameters is a json type.

It also can use “modulename.methodname” to call module’s method with the “name”.

## 2.2.4 <CALL\_EXEC/>

**summary** Call the (\*.exe) file to execute

eg.

```
<call exec_path="kill_by_port" cmd="8888"/>
<call exec_path="C:/test.exe" cmd="a b c"/>
```

**notes**

“exec\_path” is the (.exe) file’s name, if there is no ‘.exe’ in the name, the script will call the (.exe) file in the AutoXmlGo’s folder(“/bin”).

“cmd” is the file’s arguments.

## 2.2.5 <CONTINUE/>

**summary** Continues with the next iteration of the loop.

eg.

```
<for range="range(1,4)">
  <if>
    <when test="{!index}==2"><continue/></when>
  </if>
  <d_javascript>console.log('{!index}')

```

**notes** none

## 2.2.6 <COROUTINES/>

**summary** Support concurrent execution by coroutines.

eg.

```
<coroutines>
  <coroutine_func name="test1" parameters="{ 'msg': '1' }"/>
  <coroutine_func name="test2" parameters="{ 'msg': '2' }"/>
  <coroutine_func name="test3" parameters="{ 'msg': '3' }"/>
  <coroutine_func name="tM.test4" parameters="{ 'msg': '4' }"/>
  <coroutine_func name="test1" parameters="{ 'msg': '5' }"/>
  <coroutine_func name="test2" parameters="{ 'msg': '6' }"/>
  <coroutine_func name="test3" parameters="{ 'msg': '7' }"/>
  <coroutine_func name="tM.test5" parameters="{ 'msg': '8' }"/>
</coroutines>
```

**notes** none

## 2.2.7 <COROUTINE\_FUNC/>

**summary** Call the function in coroutines.

eg.

```
<coroutine_func name="test1" parameters="{ 'msg': '1' }"/>
```

**notes** none

## 2.2.8 <COROUTINE\_EXEC/>

**summary** Call the (\*.exe) file to execute in coroutines.

eg.

```
<call name="kill_by_port" cmd="8888"/>
```

**notes** none

## 2.2.9 <DEBUG/>

**summary** Write log into logs' file.

eg.

```
<debug>test1:{!test1.a}</debug>
```

**notes** none

### 2.2.10 <DOWNLOAD/>

**summary** Download the file to the local path from the url's address .

eg.

```
<download id="imgs{!index}" path="{!img_path}">{!link_url}</download>
```

**notes**

“id” is the key of the local file. if don't set “id”, the id will be the file's name.

“path” is the local path.

The tag's content(“{!link\_url}”) is the download from address.

### 2.2.11 <ELIF/>

**summary** The mean is else if clause would run.

eg. Reference *if\_tag*

**notes** none

### 2.2.12 <ELSE/>

**summary** The mean is else clause would run.

eg. Reference *if\_tag*

**notes** none

### 2.2.13 <EVAL/>

**summary** The string variable evaluated.

eg.

```
<variable>test="{ 'a':1}";</variable>
<eval id="test2">{!test}</eval>
```

**notes**

If the “id” is not setted, the variable “test” would be a json type.

If “id” is setted, the new variable “test2” would be a json type.

### 2.2.14 <F2V/>

**summary** Use file's path to be variable.

eg.

```
<f2v type="image" key="file1" as_key="file_variable"/>
```

**notes**

“type” is the space where saved in memory.(“image” is in the image memory.)

“key” is the key in selected type memory.

“as\_key” is to be variable name

## 2.2.15 <FOR/>

**summary** The for tag is used to iterate over the elements of a iterable object.

eg 1.

```
<for range="(0, 5)">
  <d_javascript>console.log('{!index}');</d_javascript>
</for>
```

**notes**

“range” is returns an iterator of integers suitable.

“{!index}” is the index of the range.

eg 2.

```
<for items="{!test_list}" var="item">
  <d_javascript>console.log('{!item}', '{!index}');</d_javascript>
</for>
```

**notes**

“items” is a iterable object.

“var” is the variable would be used in the loop.

“{!index}” is the index of the range.

## 2.2.16 <FUNC/>

**summary** Defined a function method.

eg.

```
<func name="test_func" parameters="{!a':1, 'b':[0,1,2,3]}">
  <variable>
    func_var=1;
    func_var2="2";
  </variable>
  <d_javascript>console.log('{!a}, {!b}, {!func_var}, {!func_var2}');</d_javascript>
</func>
```

**notes**

“name” is the function’s name.

“parameters” is the function’s initialization parameters.

### 2.2.17 <IF/>

#### summary

The if implement traditional control flow constructs.

<when/> is the if conditions.

<elif/> is the else if conditions.

<else/> is the else other conditions.

eg.

```
<if>
  <when test="0">
    <d_javascript>console.log('if-when');</d_javascript>
  </when>
  <elif test="{!test_if}=='123'">
    <d_javascript>console.log('if-elif');</d_javascript>
  </elif>
  <else>
    <d_javascript>console.log('if-else');</d_javascript>
  </else>
</if>
```

**notes** “test” is the conditions(type boolean).

### 2.2.18 <IMPORT/>

**summary** Import the other xml’s scripts module files into the main scripts file.

eg.

```
<import>ucar_58.xml</import>
```

#### notes

The file path is support absolute path and relative path.

### 2.2.19 <INSERT/>

**summary** Insert a item into a list.

eg.

```
<insert id="t2" index="1">"insert is ok"</insert>
```

#### notes

“id” is the list’s name.

“index” is insert index of the array.

### 2.2.20 <MODULE/>

**summary** Defined a module.

eg.

```
<module name="myModule">

  <variable>
    m_v=1;
  </variable>

  <func name="test" parameters="{ 'a':0, 'b':0 }">
    <variable>
      m_v2=2;
    </variable>
    <d_javascript>console.log('{!m_v}, {!m_v2}')

```

**notes** none

## 2.2.21 <PUT/>

**summary** Add or Update the json type variable.

eg.

```
<put id="t1" key="b">2</put>
```

**notes**

- “id” is the name of json type variable.
- “key” is the key in the json type variable.
- The tag’s content is the key’s value.

## 2.2.22 <RETURN/>

**summary** The end of the function method.

eg.

```
<func name="test" parameters="{ 'a':'' }">
  <d_javascript>console.log('func start')

```

(continues on next page)

(continued from previous page)

```
</func>
<call name="test" parameters="{ 'a': 'abc' }"/>
```

**notes** none

### 2.2.23 <SCRIPT/>

**summary** The outermost layer of the scripts's file.

eg.

```
<script name="myscript" info="some thing">
    .....
</script>
```

**notes** none

### 2.2.24 <SET/>

**summary** New or update a variable.

eg.

```
<set id="t3">"abc"</set>
```

**notes**

“id” is the variable's name.

The tag's content is the variable's new value.

### 2.2.25 <SLEEP/>

**summary** The process's sleep's times(default seconds).

eg.

```
<SLEEP>1</SLEEP>
```

**notes** none

### 2.2.26 <VARIABLE/>

**summary** Defined variables.

eg.

```
<variable>img_path='D:/screenshots/'</variable>
<d_screenshots>{!img_path}test.png</d_screenshots>
```

**notes** Support array, json, string, integer, bool and float types.

### 2.2.27 <WHEN/>

**summary** The if tag's conditions' tag.

**eg.** Reference *if\_tag*

notes

### 2.2.28 <ZIP/>

**summary** Compress files into a zip files.

eg.

```
<variable>
  testdir='D:/workspace/XmlAutoGo/examples/test';
</variable>
<zip path="{!testdir}" filename="test1" compress_path="{!testdir}"/>
<zip path="{!testdir}" filename="test2">
  [('{!testdir}', 'f1', 'nf1'), ('{!testdir}', 'f2', 'nf2'), ('{!testdir}', 'f3', 'nf3'
  →')]
</zip>
```

notes

“path” is the created file(\*.zip)’s directory.

“filename” is the created file(\*.zip)’s name.

There are two way to compress files like ↑example.

- 1.If the “compress\_path” is setted, the path’s folder will be compressed.
- 2.The tag’s content to point what the files will be compressed.

>>Tags

## 2.3 UI tags

### 2.3.1 <UI\_DLG/>

**summary** Open a dialog.

eg.

```
<ui_dlg title="test title1">this is dialog</ui_dlg>
```

**notes** “title” is the dialog’s window’s name.

### 2.3.2 <UI\_RUN\_OVER/>

**summary** Open a dialog and close the webdriver and browser.

eg.

```
<ui_run_over title="test">over!</ui_run_over>
```

**notes** “title” is the dialog’s window’s name.



>>Tags

## 2.4 Webdriver tags

Webdriver tags is the webdriver's controller.

### 2.4.1 <D\_ADD\_COOKIE/>

**summary** Add cookies.

eg.

```
<d_add_cookie>
  my='abc';
  myinfo='add by XmlAutoGo.exe';
</d_add_cookie>
```

**notes** none

### 2.4.2 <D\_CLICKABLE/>

**summary** Click the loaded element(The selector is support ['ID', 'NAME', 'CLASS\_NAME', 'XPATH', 'LINK\_TEXT', 'PARTIAL\_LINK\_TEXT', 'TAG\_NAME', 'CSS\_SELECTOR']).

eg.

```
<d_clickable timeout="10">ID=myid</d_clickable>
```

**notes** “timeout” is the times is waiting for load.(default is seconds).

### 2.4.3 <D\_DOWNLOAD/>

**summary** Download the file to the local path from the url's address .

eg.

```
<d_download id="imgs{!index}" path="{!img_path}">{!link_url}</d_download>
```

**notes**

“id” is the key of the local file. if don't set “id”, the id will be the file's name.

“path” is the local path.

The tag's content(“{!link\_url}”) is the download from address.

### 2.4.4 <D\_INPUT/>

**summary** Input the text into the text element.

eg.

```
<d_input select="id=kw" timeout="10">xmlautogo test input test</d_input>
```

**notes**

“select” is support ['ID', 'NAME', 'CLASS\_NAME', 'XPATH', 'LINK\_TEXT', 'PARTIAL\_LINK\_TEXT', 'TAG\_NAME', 'CSS\_SELECTOR']

“timeout” is the times is waiting for load.(default is seconds).

## 2.4.5 <D\_JAVASCRIPT/>

**summary** Add javascript into the browser.

eg.

```
<d_javascript>
  alert('hello word!');
</d_javascript>
```

**notes**

In the tag also support public method.

1. ADDJS\_SETVAR(key, value)

Set into the variable.

## 2.4.6 <D\_REFRESH/>

**summary** Refresh the page.

eg.

```
<d_refresh/>
```

**notes** none

## 2.4.7 <D\_SCREENSHOT/>

**summary** Screenshot the browser.

eg.

```
<d_screenshot>D:/screenshots/test.png</d_screenshot>
```

**notes** The tag's content is the image's saved path.

## 2.4.8 <D\_SLEEP/>

**summary** The process's sleep's times(default seconds).

eg.

```
<d_sleep>1</d_sleep>
```

**notes** none

### 2.4.9 <D\_SWITCH\_WIN/>

**summary** Switch the browser' window.

eg.

```
<d_switch_win>1</d_switch_win>
```

**notes** The tag's content is the number of the browser' windows(Left to Right).

### 2.4.10 <D\_TEXT\_AS\_VAR/>

**summary** Set the element text as a variable.

eg.

```
<d_text_as_var id="passcode">xpath=/html/body/div/div[2]/p[3]/b</d_text_as_var>
```

**notes**

“id” is the variable's name.

The tag's content is the element's selector.

### 2.4.11 <D\_QUIT/>

**summary** Quit the webdriver.

eg.

```
<quit/>
```

**notes** none

### 2.4.12 <D\_UPLOAD/>

**summary** Auto upload files by browser.

eg.1(one file).

```
<d_upload by="Open" path="D:/test">"test.txt"</d_upload>
```

eg.2(more than one files).

```
<d_upload by="Open" path="D:/test">["test.txt", "test1.txt"]</d_upload>
```

**notes**

“by” is the browser's upload window's name

(English is “Open”, Chinese is “开”, Japanese is “開” and so on).

“path” is the upload file's local path.

The tag's content is the local path's files name.

### 2.4.13 <D\_URL/>

**summary** Browser access address.

eg.

```
<d_url>http://www.baidu.com</d_url>
```

**notes** none

### 2.4.14 <D\_WAIT\_FOR/>

**summary** Wait for the element has been loaded(The selector is support ['ID', 'NAME', 'CLASS\_NAME', 'XPATH', 'LINK\_TEXT', 'PARTIAL\_LINK\_TEXT', 'TAG\_NAME', 'CSS\_SELECTOR']).

eg.

```
<d_wait_for timeout="10">ID=myid</d_wait_for>
```

**notes** “timeout” is the times is waiting for load.(default is seconds).

>>Tags

## 2.5 Windows OS tags

### 2.5.1 <W\_CURSOR\_CLICK\_LEFT/>

**summary** Mouse left key click once.

eg.

```
<w_cursor_click_left/>
```

**notes** none

### 2.5.2 <W\_CURSOR\_CLICK\_RIGHT/>

**summary** Mouse right key click once.

eg.

```
<w_cursor_click_right/>
```

**notes** none

### 2.5.3 <W\_CURSOR\_POS/>

**summary** Set the mouse position.

eg.

```
<w_cursor_pos x="0" y="0"/>
```

**notes** none

### 2.5.4 <W\_DIRS\_CLEAR/>

**summary** Clear the directory's files

eg.

```
<w_dirs_clear>{!file_path}</w_dirs_clear>
```

**notes** none

### 2.5.5 <W\_DIRS\_CLEAR\_ALL/>

**summary** Clear the directory's files and directories.

eg.

```
<w_dirs_clear_all>{!file_path}</w_dirs_clear_all>
```

**notes** Waring:It's a danger step.

### 2.5.6 <W\_FILE\_COPY/>

**summary** Copy from file to the new path.

eg.

```
<w_file_copy from="{!from_file_path}" to="{!to_file_path}"/>
```

**notes**

### 2.5.7 <W\_FILE\_DEL/>

**summary** Delete the file or directory.

eg.

```
<w_file_del>{!file_path}</w_file_del>
```

**notes** Waring:It's a danger step

### 2.5.8 <W\_FILE\_MOVE/>

**summary** Move the file from to.

eg.

```
<w_file_move from="{!from_file_path}" to="{!to_file_path}"/>
```

**notes** none

### 2.5.9 <W\_FILE\_READ/>

**summary** Read file's text as a variable.

eg.

```
<w_file_read id="vars">{!file_path}</w_file_read>
```

**notes**

“id” is the variable's name.

The tag's content is the read file's path.

### 2.5.10 <W\_FILE\_RENAME/>

**summary** Rename the file.

eg.

```
<w_file_rename from="{!from_file_path}" to="{!to_file_path}"/>
```

**notes** none

## CHAPTER 3

---

Examples

---





## CHAPTER 4

---

Github

---



## CHAPTER 5

---

About us

---

### 5.1 [www.freeol.cn](http://www.freeol.cn)

### 5.2 Mail

- [freeol.cn@outlook.com](mailto:freeol.cn@outlook.com)



### 6.1 version.0.0.14(Beta)

- 1|BUG|Don't show the command window when call the webdriver.(By [Freeol](#))
- 2|BUG|Fixed the XmlAutoGo project from Addjs.(By [Freeol](#))
- 3|ADD|Append Apache License 2.0 into the source.(By [Freeol](#))
- 4|MOD|Add this documents link into the "About us" window.(By [Freeol](#))
- 5|ADD|Use win32Api to control excel.(By [CQ.Jin](#))
- 6|MOD|Use win32Api to control excel.(By [CQ.Jin](#))
- 7|ADD|Add a thead to listen the excel's macro.(By [CQ.Jin](#))
- 8,9|ADD|Add a windows's control script by Autoit3.(By [CQ.Jin](#))
- 10|MOD|Support coroutines.(By [Freeol](#))
- 11|MOD|Support compress files with zip.(By [Freeol](#))
- 12|MOD|The script's variables is saved in memory.(By [Freeol](#))
- 13,14|BUG|<CALL\_EXEC/> bug fixed for get normal executable file.(By [Freeol](#))

### 6.2 version.0.0.0(Release)

- 0|ADD|The XmlAutoGo project is based on Addjs(offline project).(By [Freeol](#))