

External

This topic explains how to integrate third-party tools in the SOAtest and/or Virtualize environment.

Sections include:

- [Understanding the Definition of an External Tool](#)
- [Integrating an External \(Third-Party\) Tool](#)

Understanding the Definition of an External Tool

You can integrate any third-party (external) tool into SOAtest and/or Virtualize by creating an External tool for it, then configuring that tool appropriately. External tools are only available as transport header outputs.

Integrating an External (Third-Party) Tool

You can configure an External tool as follows:

1. Double-click the tool node.
2. Complete these fields as follows.
 - a. In the Tool's tab's **Name** field, enter the tool's name.
 - b. If you want to parameterize the tool with data source values (e.g., in the **Arguments** column), select a data source from the **Data Source** drop-down menu. This menu is only available if a data source was added to the test or Responder suite.
 - c. In the **Executable** field, enter (or browse to) the name and path of the executable file associated with this tool.
 - d. If you want to pass any flags or arguments to the tool, click the **Add** button, then click the **Modify** button and specify the Flag and Argument values in the dialog that opens. If you select a Parameterized value from a data source, each value from the specified data source column will be used as an argument.
If no flags need to be specified, the **Flag** column can be left empty



Example Using Flags and Arguments

Assume that you are on Windows and the FTP executable is called as defined in the executable field followed by all flags and values (separated by spaces).

With the executable "ftp" and the flags defined as:

Flag	Value
-cp	test.jar
test	

the final executable would be:

```
ftp -cp test.jar test
```

% arguments

If you do not specify any % arguments, the external tool will start when it is invoked, but it will not operate on any selected files, browser items, etc.

%F: This argument passes the filename and path of the selected item. It allows the tool to operate on the selected item. If you invoke a tool using this argument, the item for which it was invoked will be "ghosted" and assigned a temporary filename. You can avoid this ghosting by using the %u argument, if applicable. %F is the most commonly used argument.

%f: This argument passes the filename, *but not the path*, of the selected item.

%u: This argument passes the URL of the selected item. It works on simple URLs (URLs for pages without form submissions) as long as the associated tool can work with URLs.

%l: This argument passes any relevant line number information.

- e. If an exit value for this tool indicates the tool's success, select the **Exit value indicates success** check box. If an exit value indicates failure, leave this option off.
- f. (Optional) Set the **Output Pattern** and **Pattern Keys** options to configure how the output's format is interpreted and what the output means. If you use both options to interpret the Filename (and line number, if provided), then each time you double-click a tool message reported in the results panel, the correct file and line number will open (if your output contains file names and line numbers).
 - **Output Pattern:** Specifies how to interpret the output's format. When used in conjunction with pattern keys, this option tells SOAtest/Virtualize what the output means.
 - **Pattern Keys:** Tells SOAtest/Virtualize the line number and file name from the output pattern (in the terms of your selected source editor).

i **Output Pattern and Pattern Key Examples**

Example 1: Assume the sample tool's output is

```
1:magic one.htm
```

(where 1 is the line number, `magic_space` is an expression, and `one.htm` is the file name). In this case, you would enter the following expression in the **Output Pattern** field:

```
(.*)(:magic )(.*)
```

This expression breaks the output into three pieces (each pair of parentheses represents one piece). The first piece includes everything up until the value `:magic_space`. The second piece is the value `:magic_space`. The third piece includes everything after the value `:magic_space`.

The pattern keys settings specify how each piece will be interpreted. Here, it should be `lSF`. `l` represents line number, `s` is used as a placeholder, and `F` represents file name.

Example 2:

Assume the sample tool's output is

```
c:/home/gecko/files/a.html: (anything)
```

and the tool does not report line numbers.

You would enter `(.*)(:)(.*)` in the **Output Pattern** field and `FSS` in the **Pattern Keys** field.

- g. In the **MIME Types** field, specify which types of files this tool can work with.
- h. In the **MIME Type of Output** field, specify the type of output you want this tool to deliver.