CSV Client and CSV Call Back

This topic covers the CSV Client tool, which supports converting between CSV (delimited) and XML formats. It also addresses the CSV Call Back tool, which simplifies the handling of incoming CSV messages.

Sections include:

- About these Tools
- Working with CSV / Delimited Messages in SOAtest
- Conversion Options

About these Tools

CSV Client is an instance of the EDI Client tool that is preconfigured to operate on CSV / delimited messages. You can create a CSV Client tool directly from the Add Tool wizards. Or, you can add an EDI Client tool and change its format to CSV. The tool's name will then change to CSV Client (unless you disabled the **Use Default Name** option).

Likewise, the CSV Call Back tool is an instance of the Enhanced Call Back tool that is preconfigured to operate on CSV / delimited messages. To create a CSV Call Back tool, add an Enhanced Call Back tool and change its format to CSV. The tool's name will then change to CSV Call Back (unless you disabled the **Use Default Name** option).

Note that the CSV tools have CSV-specific conversion options; these are described in Conversion Options.

Working with CSV / Delimited Messages in SOAtest

SOAtest converts between CSV and XML format to facilitate the rapid configuration, sending, and validation of CSV messages.

If you would like to send a message with a CSV payload, you would create a CSV Client tool. Since there is no schema associated with the CSV format, you would switch to Literal view, paste in your CSV message, and then switch to Form Input view to parameterize the message.

The CSV client is designed for situations where you want to send a CSV message and you want to parameterize or script certain values of that message. If no scripting or parameterization is required, the Messaging Client can be used as an alternative. For example, assume you're using a Messaging Client to make a request that returns a CSV message. You may want to validate that message, or extract values from the message. In order to do this, you would:

1. Attach an XML Converter to the Response Output of the Messaging Client.
2. Set the format to CSV and adjust that tool's conversion options as needed.
3. Attach a XML Data Bank (to extract values) or other XML tools (to perform validations against specific values within the CSV).

Conversion Options

The following conversion options are available in CSV Client and CSV Call Back tools:

- **Separator -** Specifies which separator is used in the CSV file.
- **Quote -** Specifies which quote character is used. Quotes can be used around individual values in a CSV file. They are used in cases where the value contains the separator character that is used in the file, or in cases where a quote appears in the value (if a single quote appears in the value, you would use a double quote to surround the value; if a double quote appears in the value, you would use a single quote to surround the value). If no quotes appear, you can use either single or double quotes to surround the value.
- **Trim leading/trailing whitespace -** Specifies whether or not to strip whitespace from the beginning and end of the value.
- **First row specifies column names -** Specifies whether or not to consider the first row of the CSV file as column names. If the first row specifies column names, the elements in Form XML will show the column names as the name of each value. If the first row does not specify column names, then the elements in Form XML will use "value" as the name of each value.

These options are used converting in both directions: XML -> CSV and CSV -> XML

For example, with **First row specifies column names** set to true, a ',' separator, and double quotes as the quote character, messages would be converted as follows.

CSV:

```
Last,First
Dafoe, Daniel
"O'Hara", Maureen
```

XML:

```xml
<?xml version="1.0" encoding="Cp1252"?>
<root>
 <row>
  <Last>Dafoe</Last>
  <First>Daniel</First>
 </row>
 <row>
  <Last>O'Hara</Last>
  <First>Maureen</First>
 </row>
</root>
```