Contents

- 1 Description
- 2 Usage
- 3 Examples
- 4 See Also

Description

Exports a channel variable from the A leg to the B leg. Variables and their values will be replicated in any new channels created from the one export was called.

Usage

```
<action application="export" data="name_of_var_to_export=value"/>
```

To export multiple variables, or to export a variable without changing it's value, use Variable_export_vars

Note: if you want to override these exported channel variables in your dial-string, you have to also set the global variable *local_var_clobber* to true, for example:

```
<action application="export" data="foo=one"/>
<action application="bridge" data="{local_var_clobber=true}[foo=two]sofia/gateway/someprovider/12
```

Only now foo gets the value 'two' on your b-leg, if local_var_clobber is not set then it will keep the value 'one'.

This application may be run *inline* from the XML dialplan.

Examples

```
<action application="export" data="foo=${foo}"/>
```

This typically also sets the A leg.

```
<action application="export" data="foo=bar"/>
```

To only export to the B leg, prefix the variable name with 'nolocal:', like below:

```
<action application="export" data="nolocal:foo=bar"/>
```

It is also possible to prefix your to-be-exported channel variable with '_nolocal_', like below:

```
<action application="export" data="_nolocal_foo=bar"/>
```

Both prefixes work the same as far as FreeSWITCH is concerned.

Reason of this alternate prefix is that the 'nolocal:' variables will end up in the xml cdr of the a-leg and some xml parsers will choke on this because the nolocal: part will be regarded as namespace, which is not defined. Also, it is not possible to use ${nolocal:foo} anywhere in your dialplan because the colon (:) is used to get everything from the nth character in ${foo:n}. ${_nolocal_foo} works fine.

See Also

- Dialplan
- Variable_import