Variables

About

At it's basic level a variable simply gives a name to a piece of information, so instead of using something like 192.168.1.1 you can create a variable named local_ip_v4, now whenever you need to specify an IP address, you can refer to this variable by using ${local_ip_v4} instead of typing the actual IP address. Variable values are stored in memory on your server, when you retrieve the variable FreeSWITCH pulls the current value from memory. There are several benefits to using variables.

- Manage changes - by using a variable to define an IP address, you only specify the actual IP address once, then wherever you need the IP address you use the variable. If the IP address ever changes you just have to change it one place.
- Readability - by giving a name to a specific value it's much easier to understand the configuration, for example if you would see a number 1000 in the configuration file it's not obvious what that number means is it an extension? Is it a timeout period? You wouldn't know. You would have to spend valuable time trying to get context from the surrounding text. By using a variable with a meaningful name you can avoid the guesswork.
- Passing Data - since FreeSWITCH is built on a modular architecture, variables provide a convenient way of sharing data between different modules
- Configuration - the FreeSWITCH core and many modules have predefined variables that are used as configuration settings, these usually have a default value which you can override by setting the value of the variable. Used this way you can actually control the behavior of FreeSWITCH by changing the value of one of the predefined variables.

Global Variables

As their name implies global variables are available to the entire FreeSWITCH system and the value is the same for all channels. It's intended to be used for variables that don't change often.

Global variables can be created/modified in the configuration using the set pre-processor command. You can also set a global variable using the global_setvar API command.

The FreeSWITCH core as well as some modules have many predefined variables, some of which are set in vars.xml in the default configuration, others have a default value assigned by the FreeSWITCH core. See the Global Variables page for a full list of predefined global variables.

Channel Variables

Channel variables are variables that are specific to a single channel such as Caller-Id-Number, Dialed-Number, etc. You can create a channel variable with the set application. Because channel variables are specific to a channel, they are only available in the context of a channel such as in the Dialplan, or in scripts running as part of a dialplan.

Retrieving Variables

Variables can be retrieved using the dollar syntax ${variable_name}. Wherever you use this syntax the FreeSWITCH engine will replace it with the current value of the variable. For global variables the value will be the same for all channels. Channel variables are only available in the context of a channel and will evaluate to the value for the current channel. For example in the dialplan you can create rules based on the destination_number channel variable. As the dialplan is evaluated for a channel, it will retrieve the dialed number for this particular channel.

You can also check the current value for global variables using the eval command which you can run from the CLI.

$${variable_name} or ${variable_name}?
There is some confusion about the difference between prefixing the variable name with a single $ or a double $$. In the configuration files text using the double $$ syntax will be completely removed by the pre-processor, and assuming that there is a global variable matching the name, it will replace it with the value of the variable. If there is no matching global variable it will just remain blank. Because this is done by the pre-processor it is evaluated only when the configuration file is loaded into memory (either at startup or when reloading).

The single $ syntax is not affected by the pre-processor, instead it's evaluated at runtime, so if you use it in the dialplan, it will be evaluated again for every call. Therefore, If you change the value of a global variable, the $ syntax will reflect the new value. Another difference is that the $ syntax can be used to retrieve both global variable and channel variables, while the $$ is only for global variables.

To add to the confusion, the distinction only applies to the configuration files. In scripts, CLI, API calls etc, there is no difference between the $ and $$ syntax, both can be used to retrieve global and channel variables and are by definition evaluated at runtime.

System Defined Variables

In addition to variables you create using the set commands, there are hundreds of variables that are created by the FreeSWITCH core and the many modules that are loaded. A full list of variables is documented on the Variables Master List.