There are lots of tidbits that don't easily fall into a specific category for developers, so here are some miscellaneous and somewhat random tips that have been gleaned from conversations with the FreeSWITCH? authors.

Apache Portable Runtime

Be sure to familiarize yourself with the Apache Portable Runtime (APR) - it contains lots of handy functions, data types, objects, etc. that make FreeSWITCH? development a bit easier and more consistent across platforms. To prevent namespace collisions, apr_ objects have been mapped to the switch_ namespace. Please use the switch_ versions of apr_ objects. (Note: there is an exception to this - libmrcp). Examples:

switch_file_open in place of **apr_file_open**
switch_file_read in place of **apr_file_read**
switch_file_write in place of **apr_file_write**
switch_file_close in place of **apr_file_close**

You get the idea. See switch_apr.c for LOTS of great detail.

Channel Data

When writing your own application within FreeSWITCH? you can use switch_channel_set_private to create a hash in which you can put data specific to your own application. Example:

```
switch_channel_set_private(channel, "mydata", some_pointer_to_my_data);
...
some_pointer_to_my_data = switch_channel_get_private(channel, "mydata");
```

Debug Printing

When debug printing, consider using alternatives to raw **printf()** calls. Raw printf() calls require a STDOUT which is not always available, i.e. when running as a service under Windows or under certain circumstances when running as a daemon under Linux/Unix. Better options are available:

switch_log_printf() - the most generic way to print a debug message. Allows for debug levels and printing messages to disk files, remote apps, etc. See here and here for more information.
stream->write_function() - more memory efficient; this avoids a **malloc()** call and therefore has a lesser impact on program performance. See here for more information.

See Also: Authoring Freeswitch Modules