MetaDoc
Quick start guide

Bjørnar Grip Fjær

July 7, 2010

1 Introduction

This guide is meant to provide a quick introduction to using MetaDoc and the Meta-Doc API (MAPI). For more complete information about MetaDoc, please refer to the MetaDoc Documentation [1].

2 Needed information

Before starting to install the MetaDoc client you need some information about the MetaDoc server your client should connect to. The following information is needed:

- The host (URL) of the MetaDoc server

- The version of MetaDoc the server is running

You also need a way to securely transport your certificate to the server.

3 Retrieving the MetaDoc client

Before you start you need a version of the MetaDoc client. You can download the client from the MetaDoc GitHub repository [2]. Please make sure you download the same version as the one running on the server.

Extract the client to the wanted destination.

4 Setting up folders

The MetaDoc client uses two system folders, namely `/var/log/mapi/` and `/var/cache/mapi/` for log files and cache files, respectively. Make sure to create these folders, and make sure the user running the MetaDoc client has access to **read** and **write** to these folders.

5 Creating a X.509 certificate

MetaDoc uses X.509 certificates to identify the client on the server, and ensure that the data is safely encrypted.

The following command generates such a certificate. The value for `-subj` is not important, but dropping it will cause `openssl` to prompt for these values.

```
openssl req -x509 -newkey rsa:2048 -keyout client.key -out client.crt
-nodes -subj "/C=NO/O=example.org/CN=localhost"
```

The file `client.crt` must be sent to the server where it is bound to the site. This should be the same site name set in the configuration value **site_name** (see section 6).

6 Configuring the MetaDoc client

The MetaDoc client uses a configuration file called `metadoc.conf` in the same folder as the client itself. You can create a sample configuration file by running `mapi.py` without any parameters.

The configuration file contains the following settings:

host URI of a server responding to the MetaDoc server API [1].

key Location of file containing the private key for the client (client.key from section 5).

cert Location of file containing the certificate for the client key (client.crt from section 5).

site_name Name of the site. This must correspond to the site name set on the server for the given certificate.

ca_certs Location of file containing CA certificates that sign the key for the Meta-Doc server.

trailing_slash Whether the server uses a trailing slash on URIs, should be set to True for now.

valid Defaults to False to avoid the script from running with default configuration. Should be set to True or removed once properly configured.

Please note that file locations **must** be given as an absolute path.

The certificate file referenced by the **cert** key in the configuration must be available at the server and related to your client properly, otherwise the server will refuse the connection.

7   Using the information

Once the actions detailed above have been performed, the MetaDoc client is ready to send and receive information. However, it has to know what to do with the information it receives, and how to gather information to send.

`doc/examples/` contains examples of customized functions that produce a shadow file from user data received, project user file from project data and project quota file from received allocation data. It also contains an example of a command line tool for adding events.

Please refer to the MetaDoc documentation [1] for more information on customizing the MetaDoc client.

8   Setting up crontab

The MetaDoc client can be set up as a crontab in order to synchronize sites with the server. Below, some examples of using the client in crontab is shown. The information passed between the client and server depends on the handles passed on script execution.

A couple of useful handles are –**send-all** and –**fetch-all**, which sends all data and retrieves all data, respectively. –**all** is a combination of these. See the MetaDoc documentation for a detailed explanation of each handle [1].

```
0  0  *  *  *  /path/to/metadoc/client/mapi.py -ecs -q -l error
```

Sends event, configuration and software data to the server at midnight every night. Makes sure the script executes quietly, passing no output to `stdout`, only to `stderr` if the script execution fails. Only logs error messages.

```
0  0  *  *  0  /path/to/metadoc/client/mapi.py -upa -q -l warning
```

Fetches user, project and allocation data once a week. Makes sure the script executes quietly and logs warning messages.

9 Logging

The MetaDoc client logs to `/var/log/mapi/`, log files follow the naming schema `metadoc.client.YYYY-mm-dd.log`. The amount of information logged depends on the log level passed to **-l** on script execution. The possible log levels are **debug**, **info**, **warning**, **error** and **critical**. When **-l** is not passed, or passed with a log level not available, the level is set to **warning**. See the MetaDoc documentation for more information on logging [1].

References

[1] Fjær, Bjørnar Grip, *MetaDoc Documentation*, `http://bjornar.me/metadoc/doc.pdf`

[2] *MetaDoc Downloads*, `http://github.com/bjornarg/metadoc/downloads`