

Thursday, February 28, 2008

## **Solaris Features: Service Management Facility - Part 1: Introduction**

The Service Management Facility is a quite new feature. But sometimes i have the impression that the most used feature is the capability to use old legacy init.d scripts. But once you use SMF with all it's capabilities, you see an extremely powerful concept.

init.d

For a long time, the de-facto standard of starting up services was the init.d construct. This concept is based of startup scripts. Depending from their parametrisation they start, stop or restart a service. The definition of runlevels (what has to be started at a certain stage of booting) and the sequencing is done by linking this startup scripts in a certain directory and the naming of link.

This mechanism worked quite good, but has some disadvantages. You can't define dependencies between the services. You emulate the dependencies by sorting the links, but that's more of a kludge as a solution. Furthermore the init.d scripts run only once. When the service stops, there are no means to start it again by the system (you have to login to the system and restart it by using the init.d script directly or using other automatic methods)

With init.d a service (like httpd on port 80) is just a consequence of running scripts, not a configurable entity in itself.

Service Management Facility

SMF was invented to solve many of the problems of the old startup mechanism. Most problems from resulting from the init.d framework result from a lack of knowledge of the system about services it's running. What do i need to run my service? Is this services needed for other services? What is the status of a service? Should i restart another service (e.g. database) to circumvent problems in another service (an old web application for example)? Okay, an expert has the knowledge to do such tasks manually ... but do you want to wake up at night, just to restart this fscking old application?

The concepts of SMF enables the admin to put this knowledge into a machine readable format, thus the machine can act accordingly. This knowledge about services makes the SMF a powerful tool to manage services at your system. SMF enables the system to: starting, restarting and stopping services according to their dependenciesresulting from this the system startup is much faster, as services are started in a parallel fashion when possibleWhen a service fails, SMF restarts this servicethe delegation of tasks like starting, stopping and configuration of services to non-root usersand much moreThe following tutorial wants to give you some insights to SMF. Have fun!

Posted by Joerg Moellenkamp in English, Solaris at 07:47