Linux Standard Base Core Specification 3.1

Chapter 20. System Initialization

20.3. Comment Conventions for Init Scripts

Conforming applications may install one or more init scripts. These init scripts must be activated by invoking the **install_initd** command. Prior to package removal, the changes applied by **install_initd** must be undone by invoking **remove_initd**. See [Installation and Removal of Init Scripts](#) for more details.

install_initd and **remove_initd** determine actions to take by decoding a specially formatted block of lines in the script. This block shall be delimited by the lines

```
### BEGIN INIT INFO
### END INIT INFO
```

The delimiter lines may contain trailing whitespace, which shall be ignored. All lines inside the block shall begin with a hash character `'#'` in the first column, so the shell interprets them as comment lines which do not affect operation of the script. The lines shall be of the form:

```
# {keyword}: arg1 [arg2...]
```

with exactly one space character between the `'#'` and the keyword, with a single exception. In lines following a line containing the **Description** keyword, and until the next keyword or block ending delimiter is seen, a line where the `'#'` is followed by more than one space or a tab character shall be treated as a continuation of the previous line.

The information extracted from the block is used by the installation tool or the init-script system to assure that init scripts are run in the correct order. It is unspecified whether the information is evaluated only when **install_initd** runs, when the init scripts are executed, or both. The information extracted includes run levels, defined in [Run Levels](#), and boot facilities, defined in [Facility Names](#).

The following keywords, with their arguments, are defined:

Provides: `boot_facility_1 [boot_facility_2...]`

> boot facilities provided by this init script. When an init script is run with a **start** argument, the boot facility or facilities specified by the **Provides** keyword shall be deemed present and hence init scripts which require those boot facilities should be started later. When an init script is run with a **stop** argument, the boot facilities specified by the **Provides** keyword are deemed no longer present.

Required-Start: `boot_facility_1 [boot_facility_2...]`

> facilities which must be available during startup of this service. The init-script system should insure init scripts which provide the **Required-Start** facilities are started before starting this script.

Required-Stop: `boot_facility_1 [boot_facility_2...]`

> facilities which must be available during the shutdown of this service. The init-script system should avoid stopping init scripts which provide the **Required-Stop** facilities until this script is stopped.

Should-Start: `boot_facility_1 [boot_facility_2...]`

> facilities which, if present, should be available during startup of this service. This allows for weak dependencies which do not cause the service to fail if a facility is not available. The service may provide reduced functionality in this situation. Conforming applications should not rely on the existence of this feature.

Should-Stop: `boot_facility_1 [boot_facility_2...]`

> facilities which should be available during shutdown of this service.

Default-Start: `run_level_1 [run_level_2...]`, **Default-Stop:** `run_level_1 [run_level_2...]`

> which run levels should by default run the init script with a **start** (**stop**) argument to start (stop) the services controlled by the init script.

> For example, if a service should run in runlevels 3, 4, and 5 only, specify "Default-Start: 3 4 5" and "Default-Stop: 0 1 2 6".

Short-Description: `short_description`

> provide a brief description of the actions of the init script. Limited to a single line of text.

Description: `multiline_description`

> provide a more complete description of the actions of the init script. May span mulitple lines. In a multiline description, each continuation line shall begin with a '#' followed by tab character or a '#' followed by at least two space characters. The multiline description is terminated by the first line that does not match this criteria.

Additional keywords may be defined in future versions of this specification. Also, implementations may define local extensions by using the prefix **X-***implementor*. For example, **X-RedHat-foobardecl**, or **X-Debian-xyzzydecl**.

Example:

```
### BEGIN INIT INFO
# Provides: lsb-ourdb
# Required-Start: $local_fs $network $remote_fs
# Required-Stop: $local_fs $network $remote_fs
# Default-Start:  2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: start and stop OurDB
# Description: OurDB is a very fast and reliable database
#         engine used for illustrating init scripts
### END INIT INFO
```

The comment conventions described in this section are only required for init scripts installed by conforming applications. Conforming runtime implementations are not required to use this scheme in their system provided init scripts.

Note: This specification does not require, but is designed to allow, the development of a system which runs init scripts in parallel. Hence, enforced-serialization of scripts is avoided unless it is explicitly necessary.