

NAME

insserv - boot sequence organizer using LSB init.d script dependency information

SYNOPSIS

```
insserv [-v] [-c <config>] [-p <path>] [-d] [-f] [[/path/to/init.d/]script ...
```

```
insserv [-v] [-c <config>] [-p <path>]
[[/path/to/init.d/]script[,start=<lvl1,lvl2,...>,stop=<lvl1,lvl2,...>]
```

```
insserv [-v] [-c <config>] [-p <path>] -r [-d] [-f] [[/path/to/init.d/]script ...
```

```
insserv -h
```

DESCRIPTION

insserv is a low level tool used by **update-rc.d** which enables an installed system init script ('boot script') by reading the comment header of the script, e.g.:

```
### BEGIN INIT INFO
# Provides: boot_facility_1 [ boot_facility_2 ...]
# Required-Start: boot_facility_1 [ boot_facility_2 ...]
# Required-Stop: boot_facility_1 [ boot_facility_2 ...]
# Should-Start: boot_facility_1 [ boot_facility_2 ...]
# Should-Stop: boot_facility_1 [ boot_facility_2 ...]
# X-Start-Before: boot_facility_1 [ boot_facility_2 ...]
# X-Stop-After: boot_facility_1 [ boot_facility_2 ...]
# Default-Start: run_level_1 [ run_level_2 ...]
# Default-Stop: run_level_1 [ run_level_2 ...]
# X-Interactive: true
# Short-Description: single_line_description
# Description: multiline_description
### END INIT INFO
```

and calculating the dependencies between all scripts. It is not recommended to execute **insserv** directly unless you know exactly what you're doing, doing so may render your boot system inoperable. **update-rc.d** is the recommended interface for managing init scripts. Please be aware that the line

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

declares facilities which must be available during shutdown of the service declared in the **Provides** tag. Same holds true for

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

which declares facilities which should be available during shutdown of the service declared in the **Provides** tag. In both cases the script system should avoid stopping services which are declared by these two Stop tags until the script including these tags is stopped.

The optional **X-Interactive** keyword implies that the script using this keyword should be started alone in a concurrent boot configuration because it interact with the user at the console. Only the value 'true' is recognised. All other are ignored.

The optional **X-Start-Before** keyword implies that the script using this keyword should be started **before** the specified service names. Whereas the optional **X-Stop-After** keyword implies that the script using this keyword should be stopped **after** the specified service names. Both implies that those services now depend on the specifying script. With known dependencies and runlevel(s) **insserv** sets and reorders the corresponding symbolic links of the concerned run-levels directories.

insserv scans for **System Facilities** in the configuration file */etc/insserv.conf* and each file in the directory */etc/insserv.conf.d/*. Each line which begins with **\$** and a following name defines a system facility accordingly to the Linux Standard Base Specification (LSB). All names followed by such a system facility will declare the required dependencies of the facility. Here is an example for */etc/insserv.conf*:

```
# All local filesystems are mounted
# (done during boot phase)
$local_fs boot

# Low level networking
$network network route

# Named is operational
$named named

# All remote filesystems are mounted
# (in some cases /usr may be remote).
$remote_fs $local_fs nfs

# System logger is operational
$syslog syslog

# All network daemons are running (This was removed in LSB 1.2)
$netdaemons portmap inetd

# Services which need to be interactive
<interactive> boot.crypto
```

Names starting with a '+' sign are marked as optional. If the service with the name after the plus sign is available it will be used, if not available it is ignored silently. Words beginning with < and ending with > are keywords. Currently **<interactive>** is the only known keyword for marking a service as an interactive one, e.g. a service which requires a passphrase or password input during boot or runlevel change. The special facility **\$null** is used to enforce an empty dependency in case of **Should-Stop** and **Required-Stop**.

In addition to the defined **System Facilities** in the configuration file */etc/insserv.conf*, **insserv** also knows the special facility **\$all**. This facility indicates that a service should be inserted at the end of all services at starting and at the very beginning at stopping. Clearly all services using this facility will be grouped into one starting or stopping order.

OPTIONS

Currently there exists nine options for **insserv**.

- v, --verbose**
Write out what is currently going on.
- c <config>, --config <config>**
Specify path to the *insserv.conf* file and the *insserv.conf.d* directory. Useful for testing.
- o <path>, --override <path>**
LSB comment headers found in this path will override existing LSB comment headers of scripts in the *init.d* directory (default path is */etc/insserv/overrides/*).
- p <path>, --path <path>**
Specify path to *init.d* directory. Useful for testing.
- n, --dryrun**
Do not update symlinks.

- r, --remove**
Remove the listed scripts from all runlevels.
- d, --default**
Use default runlevels as defined in the scripts. This may restore an edited runlevel link scheme.
- f, --force**
Ignore if a required service is missed. Beside this if start and or stop levels are specified on the command line the default levels of the script will be ignored.
- u <path>, --upstart-job <path>**
Path to replace existing upstart job path. (default path is */lib/init/upstart-job*).
- s, --showall**
Output runlevel and sequence information. Do not update symlinks.
- h, --help**
Print out short usage message.

But you may use the argument syntax described in the following section.

ARGUMENTS

`[[/path/to/init.d/]`

Relative or absolute path to the init scripts base directory. This defaults to */etc/init.d/* in compliance with the LSB specification. In this case **insserv** does not add or remove a script to the runlevels declared in the script headers, but may re-order the runlevels if the order of the currently enabled scripts has changed (see option **-d**). Note that if a relative path is used **insserv** has to be called from the root directory.

`[[/path/to/init.d/]script ...`

List of scripts which have to be added to the runlevels. If a path is used it should point to the absolute or relative location of the boot scripts. **insserv c** checks for the existence of these scripts. For the runlevels the information found in the script is used.

`[[/path/to/init.d/]script[,start=<lvl1,lvl2,...>]`

List of scripts which have to be added to the specified runlevels to be started with. You may use this extension to override the default values for start and stop runlevels of the script. Note that **lvl1**, **lvl2**, ... are the known runlevels explained above. The extension `,stop=<lvl1,lvl2,...>` is also possible.

-r `[[/path/to/init.d/]script ...`

List of scripts which should be removed from the runlevels. If a path is used it should point to the absolute or relative location of the boot scripts. **insserv c** checks for the existence of these scripts.

OVERRIDES

Beside using the extensions `,start=<lvl1,lvl2,...>` and `,stop=<lvl1,lvl2,...>` it is possible to use override files replace a LSB comment header or simple provide a missing LSB comment header. This can be done by placing a file with the new LSB comment header using the same name as the boot or init script in the directory */etc/insserv/overrides/*. For third party boot scripts without LSB header it is possible to add a file with the same name in the directory */usr/share/insserv/overrides/* to make them completely LSB compliant.

UPSTART JOB COMPATIBILITY

To allow upstart jobs to work as init.d scripts, **insserv** will recognize a symlink from *path/to/init.d/script* to */lib/init/upstart-job* as upstart jobs, and instead of reading the header from the file will run the script with the argument `lsb-header` to get the script header.

EXIT CODES

The exit codes have the following conditions:

- 0 Service was successfully installed or removed
- 1 Service was not installed or removed

NOTE

Please be aware that the following patterns of boot script file names will be not accepted by **insserv**:

```
*.dpkg*
*.rpm*
*.ba*
*.old
*.new
*.org
*.orig
*.save
*.swp
*.core
*~
```

with the wildcard character `*`. Beside this all boot script file names beginning with one of the following characters

```
$.#%_+~*[]^:()~
```

will be ignored.

BUGS

Boot scripts sometimes lack a LSB comment header. Contact a package maintainer or developer of the software which provides the script to have a LSB comment header added to it.

FILES

/etc/insserv.conf

configuration file for **insserv** which defines the LSB System Facilities.

/etc/insserv.conf.d/

directory for further configuration files for declaring LSB System Facilities.

/etc/insserv/overrides/

path to replace existing LSB comment headers with the comment headers found in this path.

/etc/init.d/

path to the init script base directory as required by the Linux Standard Base Specification (LSB).

```
/etc/init.d/.depend.boot,
/etc/init.d/.depend.start,
/etc/init.d/.depend.stop
```

The [make\(1\)](#) like dependency files produced by **insserv** for *booting*, *starting*, and *stopping* with the help of [startpar\(8\)](#).

SEE ALSO

[init\(8\)](#), [startpar\(8\)](#), [update-rc.d\(8\)](#).

COPYRIGHT

2000-2009 Werner Fink,
 2009 SuSE Linux Products GmbH Nuernberg, Germany.
 2000-2003 SuSE GmbH Nuernberg, Germany,
 2007-2009 SuSE Linux Products GmbH Nuernberg, Germany.

AUTHOR

Werner Fink <feedback@suse.de>

CONTRIBUTORS

Petter Reinholdtsen
Kel Modderman