

# SMS

## Enhancements on the Client Side

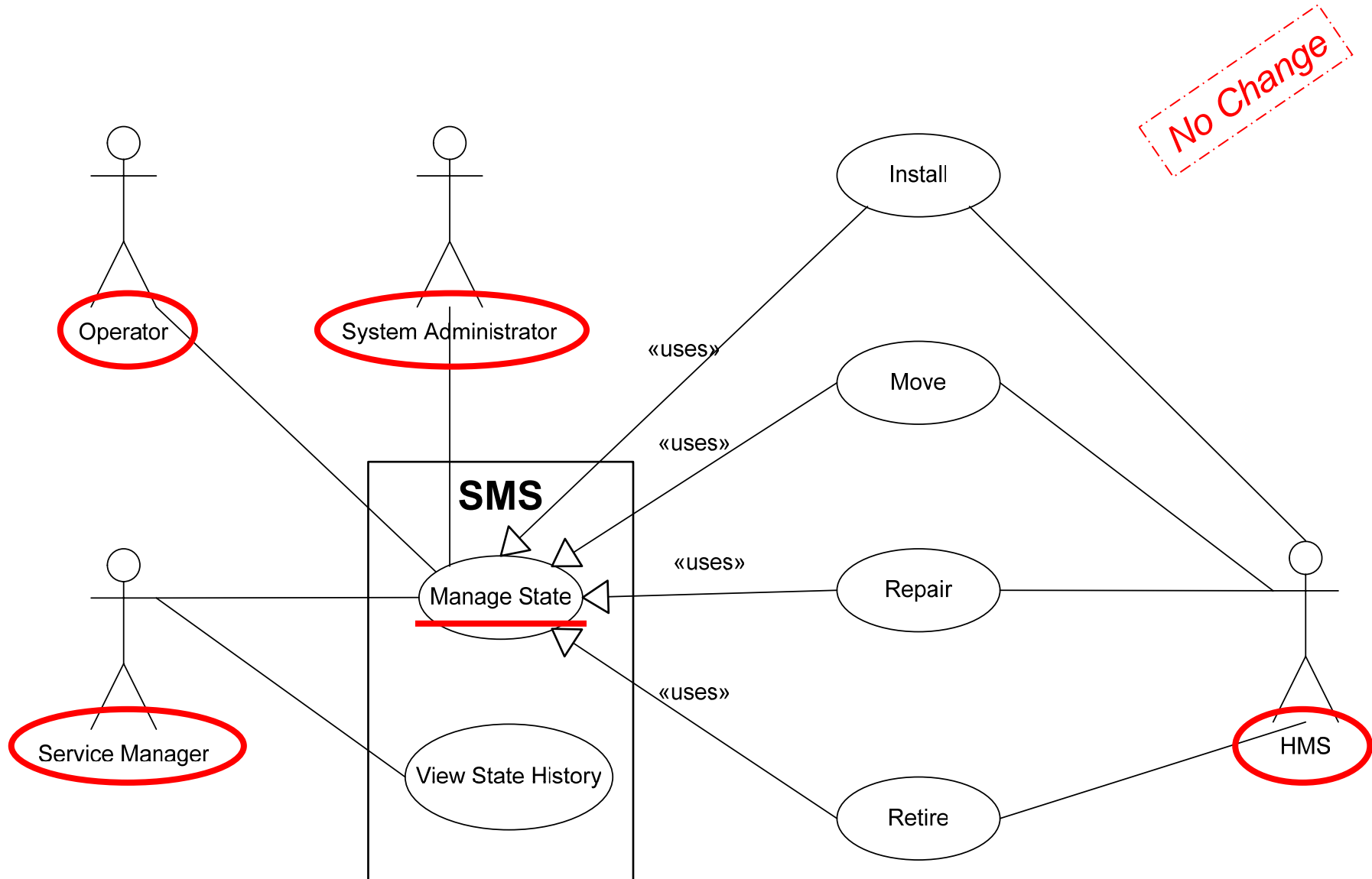
Véronique Lefébure  
ELFms Meeting  
23<sup>rd</sup> of August 2005

# Outline

- SMS Design Details
  - Reminder
  - What is new
- CVS and RPM's
- CDB Templates

# SMS Use Cases

Diagram from <http://leaf.web.cern.ch/leaf/doc/SMS-Requirements.pdf>

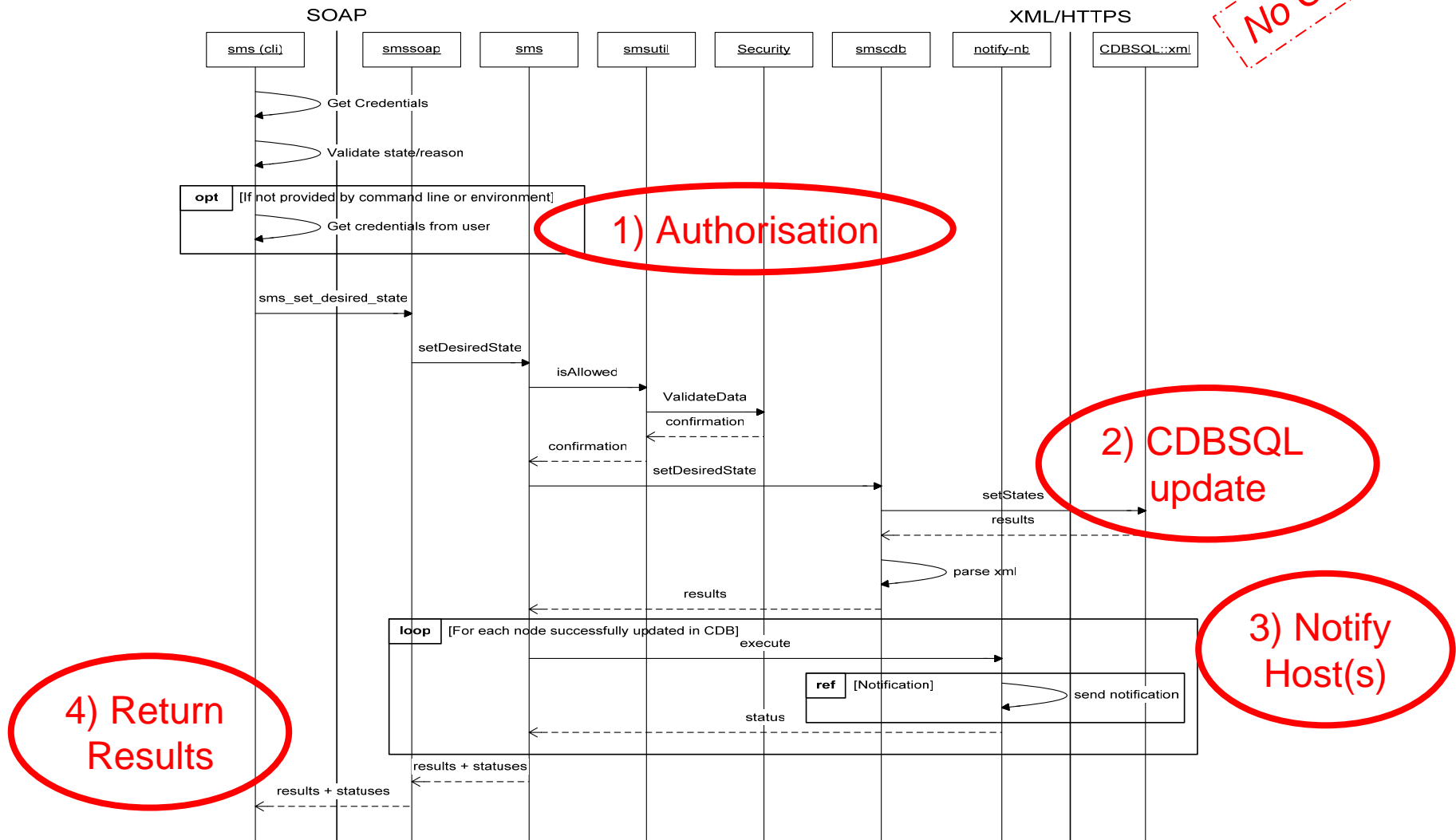


# “Set Desired State” SMS Use case

Sequence Diagram from

<http://leaf.web.cern.ch/leaf/doc/SMS-Design.pdf>

No Change



# SMS Notification

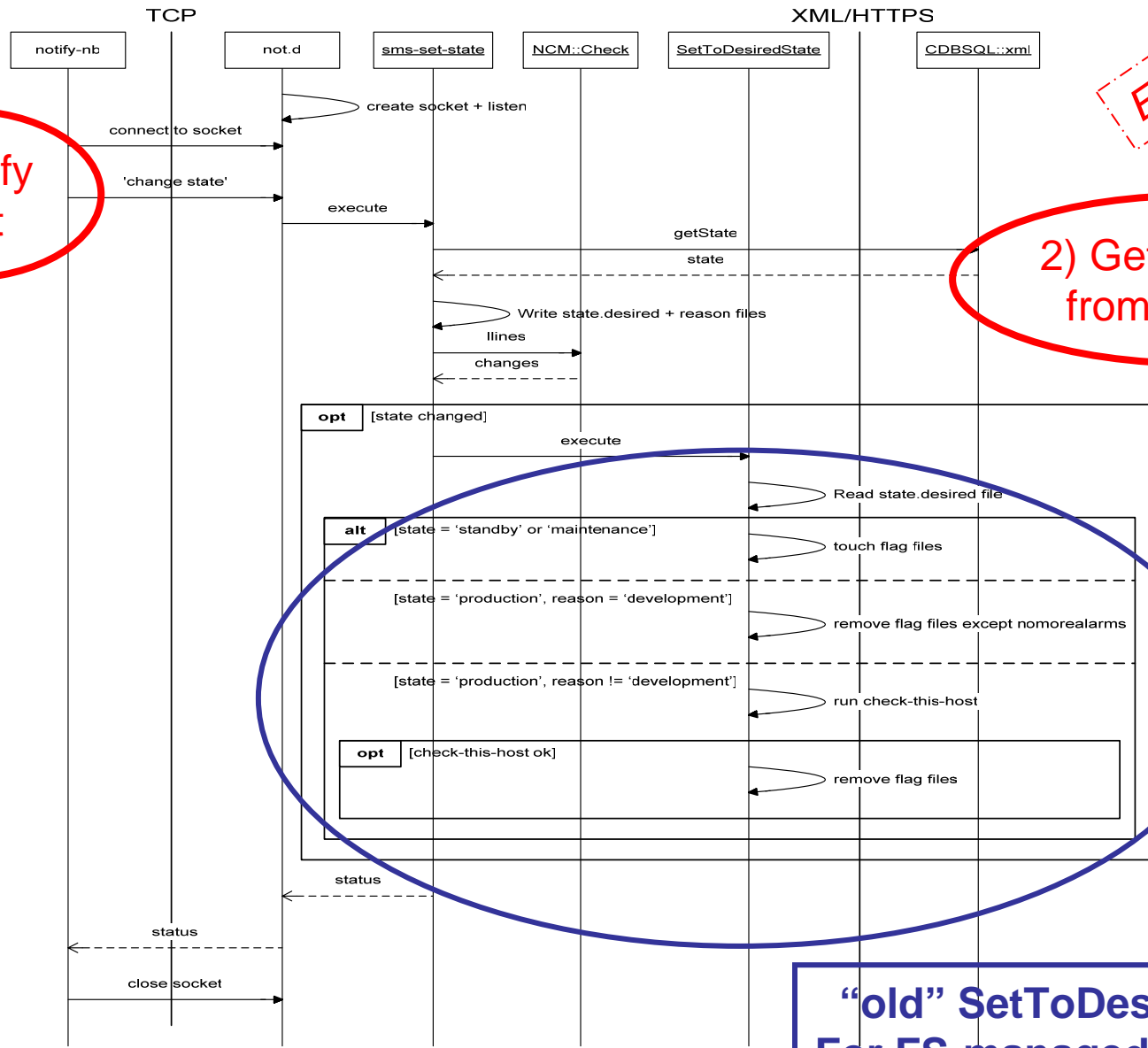
Sequence Diagram from <http://leaf.web.cern.ch/leaf/doc/SMS-Design.pdf>

Enhancements here

1) Notify Host

2) Get State Info from CDBSQL

3) Act on the Host



“old” SetToDesiredState.pl:  
For FS-managed clusters only

# What has changed (on the client side)

- Before: `sms-set-state.pl` would call `SetToDesiredState.pl` which was implemented according to FS clusters needs
- Now: `sms-set-state.pl` calls `SetToDesiredState` which in turns looks for the appropriate script to be run according to a configuration parameter defined in CDB

# How it works now, on the client:

**`/usr/sbin/sms-set-state.pl`**

1. Gets XML data from CDBSQL and creates or updates

1. `/etc/state.desired`
2. `/etc/state.desired.reasons`

2. **Runs**

```
/usr/sbin/SetToDesiredState --state=<state>  
                             --reason=<reason>
```

**which calls the first script found on the host:**

1. `/usr/sbin/SetToDesiredState.<script_ext> <state> <reason>`
2. `/usr/sbin/SetToDesiredState.<host> <state> <reason>`
3. `/usr/sbin/SetToDesiredState.<subcluster> <state> <reason>`
4. `/usr/sbin/SetToDesiredState.<cluster> <state> <reason>`
5. `/usr/sbin/SetToDesiredState.<domain> <state> <reason>`
6. `/usr/sbin/SetToDesiredState.default <state> <reason>`

# Calling Sequence

`Sms-set-state.pl`

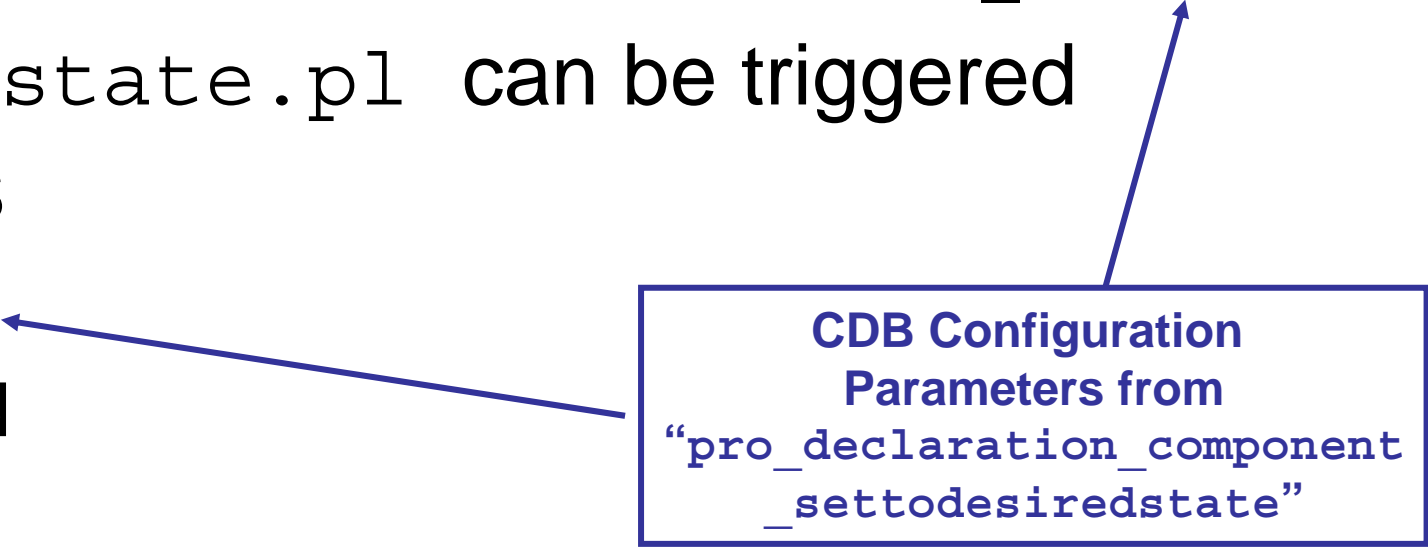
➤ `SetToDesiredState`

➤ `SetToDesiredState.<appropriate_extention>`

`Sms-set-state.pl` can be triggered

- by SMS
- at boot
- by hand

**CDB Configuration  
Parameters from  
“pro\_declaration\_component  
\_settodesiredstate”**





# Settodesiredstate NCM Component

- Reads the “/software/components/settodesiredstate” CDB configuration data
- Updates “/etc/sysconfig/settodesiredstate” :

```
RUN_AT_BOOT="no"  
SCRIPT_EXT=  
CDB_SUB_CLUSTER=  
CDB_CLUSTER="lxplus"  
CC_DOMAIN=
```

```
/usr/sbin/sms-set-state.pl --help
```

The following options are available:

- `--cfgfile <string>`  
Configuration file for this application
- `--logfile <string>`  
(default: **'/var/log/sms.log'**)  
log path/filename to use
- `--stateChanger <string>`  
(default: `'/usr/sbin/SetToDesiredState'`)  
Script that will act on the host
- `--statefile <string>`  
(default: `'/etc/state.desired'`)  
Where to store the desired state locally on the host
- `--statereasonfile <string>`  
(default: `'/etc/state.desired.reasons'`)  
Where to store the state reason(s) locally on the host
- `--urlroot <string>`  
(default: `'http://oraweb.cern.ch:9000/pls/cdbsql/xml.getstate'`)  
Where to get the state and reasons from

```
2005/08/22-11:56:44 \  
[INFO] /usr/sbin/sms-set-state.pl Invoked  
2005/08/22-11:56:44 \  
[INFO] Running /usr/sbin/SetToDesiredState \  
--state=production  
2005/08/22-11:56:49 \  
[INFO] State set to "production" completed.
```

```
/usr/sbin/SetToDesiredState --help
```

The following options are available:

- cfgfile <string>  
Configuration file for this application
- logfile <string>  
(default: `'/var/log/SetToDesiredState.log'`)  
log path/filename to use
- reason <string>  
input reason for state
- state <string>  
input desired state

# “SetToDesiredState.default”

```
#!/bin/bash
#
# SetToDesiredState Default
#
STATE=$1
REASON=$2
case $STATE in
  production)
    echo "$0: State production: remove /etc/nologin";
    rm -f /etc/nologin;
    ;;

  maintenance)
    echo "$0: State maintenance: doing nothing";
    ;;

  standby)
    echo "$0: State standby: doing nothing";
    ;;

  *)
    echo "$0: Unknown state \"$STATE\"."
    ;;
esac
```

# CVS Code and RPM for sms-set-state

- [fio]/fabric/sms-set-state/SRC
  - [SetToDesiredState](#)
  - [SetToDesiredState.default](#)
  - [settodesiredstate.sysconfig](#)
  - [sms-set-state.pl](#)
  - [sms-set-state.rc](#)
  - [sms.logrotate](#)
  - [sms.notd](#)
- [fio]/fabric/sms-set-state/RPM
  - rpm -ql CERN-CC-sms-set-state :
    - /etc/logrotate.d/sms-set-state
    - /etc/not.d
    - /etc/not.d/sms-set-state
    - /etc/rc.d/init.d/sms-set-state
    - /etc/sysconfig/settodesiredstate
    - /usr/sbin/SetToDesiredState
    - /usr/sbin/SetToDesiredState.default
    - /usr/sbin/sms-set-state.pl

# CVS Code and RPM for ncm-settodesiredstate

- [elfms]/elfms/quattor/ncm-components/cern-it-fio/ncm-settodesiredstate:
  - [ChangeLog](#)
  - [LICENSE](#)
  - [MAINTAINER](#)
  - [Makefile](#)
  - [README](#)
  - [config.mk](#)
  - [pro\\_declaration\\_component\\_settodesiredstate.tpl.cin](#)
  - [settodesiredstate.pm.cin](#)
  - [settodesiredstate.pod.cin](#)
- rpm -ql ncm-settodesiredstate
  - /usr/lib/ncm/config/settodesiredstate
  - /usr/lib/perl/NCM/Component/settodesiredstate.pm
  - /usr/share/doc/ncm-settodesiredstate-1.0.9
  - /usr/share/doc/ncm-settodesiredstate-1.0.9/ChangeLog
  - /usr/share/doc/ncm-settodesiredstate-1.0.9/LICENSE
  - /usr/share/doc/ncm-settodesiredstate-1.0.9/MAINTAINER
  - /usr/share/doc/ncm-settodesiredstate-1.0.9/README
  - /usr/share/doc/ncm-settodesiredstate-1.0.9/settodesiredstate.pod
  - /usr/share/doc/pan-templates/components/pro\_declaration\_component\_settodesiredstate.tpl
  - /usr/share/man/man8/ncm-settodesiredstate.8.gz

# SetToDesiredState Scripts for FS clusters

- [fio]/fabric/settodesiredstate-FS/SRC:
  - [SetToDesiredState.lsfmaster](#)
  - [SetToDesiredState.lxbatch](#)
  - [SetToDesiredState.lxbuild](#)
  - [SetToDesiredState.lxdev](#)
  - [SetToDesiredState.lxgate](#)
  - [SetToDesiredState.lxplus](#)
  - [SetToDesiredState.lxserv](#)
- All stored in one RPM “CERN-CC-settodesiredstate-FS”

# CDB Templates

- ***pro\_declaration\_component\_settodesiredstate.tpl :***

```
declaration template pro_declaration_component_settodesiredstate;
include pro\_declaration\_component\_type;
define type component_settodesiredstate_type = {
    include component_type
    "run_at_boottime" : string with match (self, 'yes|no')
    "script_ext" ? string
};
type "/software/components/settodesiredstate"
    = component_settodesiredstate_type;
```

- ***pro\_software\_packages\_state.tpl:***

```
template pro_software_packages_state;
"/software/packages"=pkg_add("CERN-CC-sms-set-state", "3.2-6"
                                , "noarch");
```

- ***In pro\_software\_ixplus\_slc3.tpl:***

```
include pro_software_packages_state;
"/software/packages"=pkg_add("CERN-CC-settodesiredstate-FS", "1.0-2"
                                , "noarch");
```



# More ?

- Authors/Maintainers/Responsibles are Thorsten and Veronique
- Feedback, wishes, ...are welcome