

Node State Manager

Part of the Lifecycle subsystem in GENIVI

About

The Node State Manager (NSM) is the central functional component that gathers information on the current running state of the embedded system. The NSM component provides a common implementation framework for the system state machine. It collates information from multiple sources and uses the data to determine the current state(s). Based on the given data, the NSM notifies registered consumers (applications or other platform components) of relevant changes in system state. Node state information can also be requested on-demand via provided D-Bus interfaces. The node state manager also provides shutdown management including shutdown request notification to consumers. The node state management is the last/highest level of escalation on the node and will issue the reset instruction and supply control logic. It is notified of errors and other status signals from components that are responsible for monitoring system health.

Internally, node state management is made up of a common generic component, Node State Manager (NSM), and a system-specific state machine (NSMC) that is plugged into the Node State Manager. Through this architecture there can be a standardized solution with stable interfaces towards the applications, which still allows for product-specific behavior through the definition of the specific state machine.

Building NSM

```
./autogen.sh
./configure
make
make install
```

To run the test:

```
Overwrite the "libNodeStateMachine.so" with the test version:
cp /usr/lib/libNodeStateMachine.so /usr/lib/libNodeStateMachine.so_org
cp /usr/lib/libNodeStateMachineTest.so /usr/lib/libNodeStateMachine.so
```

```
Create an own dbus instance:
dbus-launch
```

```
The command will print a DBUS_SESSION_BUS_ADDRESS. Copy/paste and
export it as system bus:
export DBUS_SYSTEM_BUS_ADDRESS=unix:path=...
```

```
Start the NodeStateManager (in background)
NodeStateManager &
```

```
Run the test:
NodeStateTest
```

See also

[Node Startup Controller](#)

[Node Health Monitor](#)