Daemons

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clexecd</td>
<td>This is used by cluster kernel threads to execute userland commands (such as the run_reserve and dofsck commands). It is also used to run cluster commands remotely (like the cluster shutdown command). This daemon registers with failfastd so that a failfast device driver will panic the kernel if this daemon is killed and not restarted in 30 seconds.</td>
</tr>
<tr>
<td>cl_ccrad</td>
<td>This daemon provides access from userland management applications to the CCR. It is automatically restarted if it is stopped.</td>
</tr>
<tr>
<td>cl_eventd</td>
<td>The cluster event daemon registers and forwards cluster events (such as nodes entering and leaving the cluster). There is also a protocol whereby user applications can register themselves to receive cluster events. The daemon is automatically respawned if it is killed.</td>
</tr>
<tr>
<td>cl_eventlogd</td>
<td>Cluster event log daemon logs cluster events into a binary log file. At the time of writing for this course, there is no published interface to this log. It is automatically restarted if it is stopped.</td>
</tr>
<tr>
<td>failfastd</td>
<td>This daemon is the failfast proxy server. The failfast daemon allows the kernel to panic if certain essential daemons have failed.</td>
</tr>
<tr>
<td>rgmd</td>
<td>The resource group management daemon which manages the state of all cluster-unaware applications. A failfast driver panics the kernel if this daemon is killed and not restarted in 30 seconds.</td>
</tr>
<tr>
<td>rpc.fed</td>
<td>This is the fork-and-exec daemon, which handles requests from rgmd to spawn methods for specific data services. A failfast driver panics the kernel if this daemon is killed and not restarted in 30 seconds.</td>
</tr>
<tr>
<td>rpc.pmfd</td>
<td>This is the process monitoring facility. It is used as a general mechanism to initiate restarts and failure action scripts for some cluster framework daemons (in Solaris 9 OS), and for most application daemons and application fault monitors (in Solaris 9 and 10 OS). A failfast driver panics the kernel if this daemon is stopped and not restarted in 30 seconds.</td>
</tr>
<tr>
<td>pnmd</td>
<td>Public management network service daemon manages network status information received from the local IPMP daemon running on each node and facilitates application failovers caused by complete public network failures on nodes. It is automatically restarted if it is stopped.</td>
</tr>
<tr>
<td>scdpmd</td>
<td>Disk path monitoring daemon monitors the status of disk paths, so that they can be reported in the output of the cldev status command. It is automatically restarted if it is stopped.</td>
</tr>
</tbody>
</table>

File locations

<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>man pages</td>
<td>/usr/cluster/man</td>
</tr>
<tr>
<td>log files</td>
<td>/var/cluster/logs /var/adm/messages</td>
</tr>
<tr>
<td>sccheck logs</td>
<td>/var/cluster/sccheck/report.&lt;date&gt;</td>
</tr>
<tr>
<td>CCR files</td>
<td>/etc/cluster/CCR</td>
</tr>
<tr>
<td>Cluster infrastructure file</td>
<td>/etc/cluster/CCR/infrastructure</td>
</tr>
</tbody>
</table>

SCSI Reservations

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display reservation keys</td>
<td><code>scsi2:/usr/cluster/lib/sc/pgre -c pgre_inkeys -d /dev/disk/d4s2</code>&lt;br&gt;<code>scsi3:/usr/cluster/lib/sc/scsi -c inkeys -d /dev/disk/d4s2</code></td>
</tr>
<tr>
<td>determine the device owner</td>
<td><code>scsi2:/usr/cluster/lib/sc/pgre -c pgre_inresv -d /dev/disk/d4s2</code>&lt;br&gt;<code>scsi3:/usr/cluster/lib/sc/scsi -c inresv -d /dev/disk/d4s2</code></td>
</tr>
</tbody>
</table>

Cluster information

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quorum info</td>
<td><code>scstat -q</code></td>
</tr>
<tr>
<td>Cluster components</td>
<td><code>scstat -pv</code></td>
</tr>
<tr>
<td>Resource/Resource group status</td>
<td><code>scstat -g</code></td>
</tr>
<tr>
<td>IP Networking Multipathing</td>
<td><code>scstat -l</code></td>
</tr>
</tbody>
</table>
Cluster Configuration

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrity check</td>
<td><code>sccheck</code></td>
</tr>
<tr>
<td>Configure the cluster (add nodes, add data services, etc)</td>
<td><code>scinstall</code></td>
</tr>
<tr>
<td>Cluster configuration utility (quorum, data services, resource groups, etc)</td>
<td><code>scsetup</code></td>
</tr>
<tr>
<td>Add a node</td>
<td><code>scconf -a -T node=&lt;host&gt;</code></td>
</tr>
<tr>
<td>Remove a node</td>
<td><code>scconf -r -T node=&lt;host&gt;</code></td>
</tr>
<tr>
<td>Prevent new nodes from entering</td>
<td><code>scconf -a -T node=</code></td>
</tr>
<tr>
<td>Put a node into maintenance state</td>
<td><code>scconf -c -q node=&lt;node&gt;,maintstate</code></td>
</tr>
<tr>
<td>Get a node out of maintenance state</td>
<td><code>scconf -c -q node=&lt;node&gt;,reset</code></td>
</tr>
<tr>
<td>Adding a device to the quorum</td>
<td><code>scconf -a -q globaldev=d11</code></td>
</tr>
<tr>
<td>Removing a device to the quorum</td>
<td><code>scconf -r -q globaldev=d11</code></td>
</tr>
</tbody>
</table>
| Remove the last quorum device | `Evacuate all nodes`  
`#scconf -c -q installmode`  
`#remove the quorum device`  
`#scstat -q` |
| Resetting quorum info | `scconf -c -q reset` |
| Bring a quorum device into maintenance mode | `obtain the device number`  
`#scdidadm -L`  
`#scconf -c -q globaldev=<device>,maintstate` |
| Bring a quorum device out of maintenance mode | `scconf -c -q globaldev=<device>,reset` |

Admin Quorum Device

Quorum devices are nodes and disk devices, so the total quorum will be all nodes and devices added together. You can use the scsetup GUI interface to add/remove quorum devices or use the below commands.

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
</table>
| Adding a device to the quorum | `scconf -a -q globaldev=d11`  
Note: if you get the error message "unable to scrub device" use `scgdevs` to add device to the global device namespace. |
| Removing a device to the quorum | `scconf -r -q globaldev=d11` |
| Remove the last quorum device | `Evacuate all nodes`  
`#scconf -c -q installmode`  
`#remove the quorum device`  
`#scstat -q` |
| Resetting quorum info | `scconf -c -q reset`  
Note: this will bring all offline quorum devices online |
| Bring a quorum device into maintenance mode | `obtain the device number`  
`#scdidadm -L`  
`#scconf -c -q globaldev=<device>,maintstate` |
| Bring a quorum device out of maintenance mode | `scconf -c -q globaldev=<device>,reset` |

Device Configuration

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lists all the configured devices including paths across all nodes.</td>
<td><code>scdidadm -L</code></td>
</tr>
<tr>
<td>List all the configured devices including paths on node only.</td>
<td><code>scdidadm -l</code></td>
</tr>
<tr>
<td>Reconfigure the device database, creating new instances numbers if required.</td>
<td><code>scdidadm -r</code></td>
</tr>
</tbody>
</table>
| Perform the repair procedure for a particular path (use then when a disk gets replaced) | `scdidadm -R <c0t0d0s0> - device`  
`scdidadm -R 2 - device`  
`scdidadm -R 2 - device id` |
| Configure the global device namespace | `scgdevs` |
| Status of all disk paths | `scdpm -p all:all`  
Note: (<host>:<disk>) |
| Monitor device path | `scdpm -m <node:disk path>` |

http://www.datadisk.co.uk/html_docs/sun/sun_cluster_31_cs.htm 24/10/2008
Unmonitor device path

```
scdpm –u <node:disk path>
```

### Disks group

#### Adding/Registering

```
sconf -a -D type=vxvm,name=appdg,nodelist=<host>:<host>,preferred=treue
```

#### Removing

```
sconf -r -D name=<disk group>
```

#### adding single node

```
sconf -a -D type=vxvm,name=appdg,nodelist=<host>
```

#### Removing single node

```
sconf -r -D name=<disk group>,nodelist=<host>
```

#### Switch

```
sswitch -z -D <disk group> -h <host>
```

#### Put into maintenance mode

```
sswitch -m -D <disk group>
```

#### take out of maintenance mode

```
sswitch -z -D <disk group> -h <host>
```

#### onlining a disk group

```
sswitch -Z -D <disk group> -h <host>
```

#### offlining a disk group

```
sswitch -F -D <disk group>
```

#### Resync a disk group

```
sconf -c -D name=appdg,sync
```

### Transport cable

#### Enable

```
sconf –c –m endpoint=<host>:qfe1,state=enabled
```

#### Disable

```
sconf –c –m endpoint=<host>:qfe1,state=disabled
```

#### Note: it gets deleted

### Resource Groups

#### Adding

```
scrgadm -a -g <res_group> -h <host>,<host>
```

#### Removing

```
scrgadm –r –g <group>
```

#### changing properties

```
scrgadm -c -g <resource group> -y <property=value>
```

#### Listing

```
scstat -g
```

#### Detailed List

```
scrgadm -pv -g <res_group>
```

#### Display mode type (failover or scalable)

```
scrgadm -pv -g <res_group> | grep 'Res Group mode'
```

#### Offlining

```
sswitch -F -g <res_group>
```

#### Onlining

```
sswitch -Z -g <res_group>
```

#### Unmanaging

```
sswitch –u –g <res_group>
```

#### Managing

```
sswitch –o –g <res_group>
```

#### Switching

```
sswitch –z –g <res_group> –h <host>
```

#### Adding failover network resource

```
scregadm -a -L -g <res_group> -l <logicalhost>
```

#### Adding shared network resource

```
scregadm -a -S -g <res_group> -l <logicalhost>
```

#### adding a failover apache application and attaching the network resource

```
scregadm -a -j apache_res -g <res_group> \
  -t SUNW.apache -y Network_resources_used = <logicalhost> \n  -y Scalable=False -y Port_list = 80/tcp \n  -x Bin_dir = /usr/apache/bin
```

#### adding a shared apache application and attaching the network resource

```
scregadm -a -j apache_res -g <res_group> \n  -t SUNW.apache -y Network_resources_used = <logicalhost> \n  -y Scalable=True -y Port_list = 80/tcp \n  -x Bin_dir = /usr/apache/bin
```

#### Create a HAStoragePlus failover resource

```
scregadm -a -g rg_oracle -j hasp_data01 -t SUNW.HAStoragePlus \n  -x FileSystemMountPoints=/oracle/data01 \n  -x Affinityon=true
```

#### Removing

```
scregadm –r –j res-<ip>
```

#### Note: must disable the resource first

#### changing properties

```
scrgadm -c -j <resource> -y <property=value>
```

#### List

```
scstat -g
```

#### Detailed List

```
scrgadm -pv -j <resource>
```

#### Disable resource monitor

```
scrgadm -e -M -j res-<ip>
```

#### Enable resource monitor

```
scrgadm -e -W -j res-<ip>
```

#### Disabling

```
sswitch -n -j res-<ip>
```

#### Enabling

```
scswitch -e -j res-<ip>
```

#### Clearing a failed resource

```
scswitch –c –h<host>,<host> -j <resource> -f STOP_FAILED
```

---

http://www.datadisk.co.uk/html_docs/sun/sun_cluster_31_cs.htm

24/10/2008
### Find the network of a resource

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>`scrgadm -pvv -j &lt;resource&gt;</td>
</tr>
</tbody>
</table>

### Removing a resource and resource group

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>scswitch -F -g rgroup-1</code></td>
</tr>
<tr>
<td><code>scrgadm -r -j res-ip</code></td>
</tr>
<tr>
<td><code>scrgadm -r -g rgroup-1</code></td>
</tr>
</tbody>
</table>

### Resource Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding</td>
<td><code>scrgadm -a -t &lt;resource type&gt;</code> i.e SUNW.RAStoragePlus</td>
</tr>
<tr>
<td>Deleting</td>
<td><code>scrgadm -r -t &lt;resource type&gt;</code></td>
</tr>
<tr>
<td>Listing</td>
<td>`scrgadm -pv</td>
</tr>
</tbody>
</table>