# Administrator's Guide
for SaberNet DCS 2.0

Title:	SaberNet DCS Administrator's Guide
Author:	Seth Remington <sremington@saberlogic.com>
Author:	Matthew Ranostay <mranostay@saberlogic.com>
Date:	2006-08-16
Revision:	1.8
Description:	Documentation for properly administrating an enterprise SaberNet DCS deployment.

Contents

Pyro Name Server and Event Server

PYRO consists of two parts, a server side, and client side. The server side is made up of an 'Event Server' and a 'Name Server'. The client side is of an application(i.e like 'sndcs_gtk'), which PYRO refers to these as 'Subscribers'. The 'Name Server' listens on a subnet/network segment for an 'Event Server' to register to it. It is import to note that ONLY ONE 'Name Server' can run on subnet at any given time, so chose wisely which host runs it. One or more Event Servers can be run on the same network segment, usually one per namespace(NOTE: See below for reference). Applications then can subscribe to the namespace(s) via the 'Name Server' using network broadcasts or a direct connection. When an client sends a event to the namespace, the 'Event Server' pushes it to all the clients. The events can be any object, except sockets and file descriptors.

For more in-depth discussion on PYRO, visit http://pyro.sourceforge.net/manual/PyroManual.html

SaberNet DCS requires an instance of the Pyro name server and event server to be running somewhere on the network. For a basic installation it is easiest to allow the SaberNet server (sndcsd) to automatically start an instance of the name and event servers, which it will do by default. In certain circumstances it may be desirable to run a separate instance of the name and event server. For instance, if you are running other Pyro based programs you would probably want to be running a separate shared instance of the name and event server. Another good reason to run a separate instance is that otherwise, if the sndcsd daemon is restarted all of the clients will need to be restarted as well because the heartbeat would be lost if the name/event server was restarted.

To automatically start the name/event server answer 'Y' to the following prompt during the configuration:

```
Do you want to automatically start the pyro Name Server? [Y/n]:
```

If you answer 'N' you will be prompted to specify the host that the name/event server is running on:

```
What host is the Pyro Name Server running on (empty string means use broadcast only)?
```

Enter the host that the name server is running on. Alternatively you can leave it blank and just hit <Enter>. This will cause Pyro to broadcast for the name server. This has certain advantages but also add a bit of overhead as well.

Logging

The logging configuration is in '/etc/sndcs/logging.conf' for POSIX systems, or for Windows systems its sys.prefix + '\etc\sndcs\logging.conf' *(NOTE: Usually C:\Python24 unless changed when installed).* To enable/disable a logging function change the 'keys' parameter under the 'formatters' section. Also to change the debuging level for a function, first find its section and then change the 'level' parameter. The possible choices in order of the most verbose output are DEBUG **(NOTE: Use this if you plan to send in a bug report)**, INFO, WARNING, ERROR, and CRITICAL.

Namespace

PYRO allows multiple namespaces, so you could have multiple instances of DCS pointing at different databases *(i.e More than one company shares a network).* By default DCS uses "sndcs" as it default namespace. If you wish to change the namespace please do the following:

1. Open the 'sndcsd.conf' file located in '/etc/sndcs' for POSIX systems, or for Windows in sys.prefix + '\etc\sndcs\'.

2. Edit the 'namespace' parameter to what you want, as long as it's not conflicting with another namespace.

3. Edit all the clients 'sndcs2.conf' files *(NOTE: Located in the same directory as above)* 'namespace' parameter to point to one defined above.

Plugins

For information about plugins please refer to the Plugin Guide

Barcode Format

SaberNet DCS attempts to be as efficient as possible when in barcode mode. Because of this the barcodes themselves need to contain more than just the basic data. There should also be control characters suronding the data. An example will make it clear:

```
$305|s$   <----- barcode data
| | |||
| | ||--> Main selection signifier indicating the end of the barcode
| | |---> The 's' is the clock in/out and selection signifier
| | ----> The '|' is the separator signifier... it separated the do-
main specific data (the employee number in this case) from the ac-
tion data (select employee in this case)
| ------> This is the employee's number (305)
--------> Main selection signifier indicating the start of the barcode
```

The following table defines all of the available signifiers. The signifiers will be configurable in a future version of SaberNet DCS but for now they are hardcoded.

Main Signifiers (these begin and end the barcode)	
Code	**Descroption**
$	Selection
*	Action

Separator Signifier (separates data from commands)	
Code	**Description**
|	Separator Signifier

Selection Signifiers	
Code	**Description**
s	Clock in/out and selection
l	Lunch in/out
b	Break in/out
r	Resume
o	Clock out

Action Signifiers	
Code	**Description**
i	Indirect activity
p	Production activity
s	Setup Activity

Commands

SaberNet DCS has several keyboard commands. All commands start with the forward slash '/' followed by the command. The available commands are:

Command	**Description**
quit	End the application

Command	**Description**
exit	End the application
about	Show the about box