# Installation Guide
for SaberNet DCS 2.0

Title:	SaberNet DCS Installation Guide
Author:	Seth Remington <sremington@saberlogic.com>
Author:	Matthew Ranostay <mranostay@saberlogic.com>
Date:	2014-09-15
Revision:	1.8.2.8
Description:	Documentation for installing SaberNet DCS from source.

Contents

Introduction

This installation guide will describe how to install SaberNet DCS from source. There are other options than installing from source. If you are running on a win32 platform there is an all-in-one installer for the client and for the server.

SaberNet DCS is a client/server application. Both the client and server are installed from the same source code but they have separate dependencies and installation procedures. The instructions below will specify when certain instructions apply to only the client, the server, or both.

In addition there may be some differences between installing on Linux and win32. Those differences will be noted as well. Otherwise assume the instructions apply to all architectures.

The SaberNet DCS server is named sndcsd (SaberNet DCS Daemon). There are several clients (and more planned) but for right now the most mature is the GTK+ client for Linux and win32. It is named sndcs_gtk.

Dependencies

Both

- Python >= 2.3 (http://www.python.org)
- Pyro >= 3.5; < 4.0 (https://pypi.python.org/pypi/Pyro) NOTE: Pyro 4 will not work as it does not yet include the Event Server functionality.

Client

- PyGTK >= 2.4 (http://www.pygtk.org) including libglade support
- GTK+ Runtime Environment
- For spellcheck support you will also need Python gtkspell support installed (i.e. python-gnome2-extras or python-gtkspell)

Mac OS X

- XDarwin (http://xdarwin.org)
- XCode >= 2.0 (http://www.apple.com/macosx/features/xcode)

Server

- Webware >= 0.9 (http://www.webwareforpython.org) NOTE: if you are using Python 2.5 you will need to use Webware >= 0.9.3 due to fixes in Webware's internal ImportSpy code.
- A RDBMS supported by Webware/MiddleKit and it's associated Python wrapper (i.e. MySQL/MySQLdb, PostgreSQL/psycopg, MSSQL/mx.ODBC) **NOTE: PostgreSQL and MSSQL support are experimental. For a production environment MySQL is the only fully tested RDBMS platform.**

If you are using Microsoft SQL Server as your database backend you will also require the following dependencies to be installed: * mxBase (http://www.egenix.org) * mxODBC (http://www.egenix.org)

Note: the installation uses the builtin Python distutils module. Many Linux distributions split this off into a python-dev (or similarly named) package. Please make sure you have that installed before proceeding or you may get an error when you run "python setup.py install"

Installation

1. Untar sndcs-<version>.tar.gz
2. cd sndcs-<version>
3. ./configure_sndcs.py (Please be sure to read and understand the Configuration and Databases section before you proceed with this step.)
4. python setup.py install (as root) --or-- run configure_sndcs as root and choose to install when prompted
5. Start the DCS Server Daemon (Please see the Administrator's Guide for instructions.)
6. Start the DCS Web Server (Please see the Administrator's Guide for instructions.)
7. You'll probably want to populate the database with some base data. See Base Data for instructions on how to do this.
8. Run your frontend of choice (i.e. sndcs_gtk, sndcs_curses, etc...)

Configuration

The configuration files for the server and clients are initially created by running the configure_sndcs.py script. This is a simple command line program that will prompt you for configuration parameters. Most prompts will end with a default answer and/or available answers to choose from within square brackets [...]. For example, [y,N] means that the available answers are y(es) and n(o) and the capitalized 'N' means that no is the default which will be selected if you just press <Enter>. If configure_sndcs.py has been run previously the defaults will be read in from your previous configuration so you can simply <Enter> through the prompts that you don't want to change and it will keep your old configuration.

When running the configure_sndcs.py script you will first be prompted:

```
Do you want to install the client, server, or both? [c/s/b]:
```

Selecting c(lient) will configure the clients and s(erver) the server. Selection b(oth) will simply run the server configuration followed by the client one. *It is important to note that the actual installation will only install the pieces for which there is a configuration file for.* So if you only configure the client, the server (sndcsd) will *NOT* be installed.

Server Configuration

Section	Parameter	Description
webware	webware_path	Where is Webware installed? What is the path to Webware's installation directory. If you installed Webware from a deb package it is probably in /usr/share/webware. A check will be performed to make sure that the path is indeed a valid installation of Webware so don't worry if you are wrong because I will let you know as soon as you hit OK.
database	engine	Select your RDBMS engine. Which database will you be using as you backend data store? Available choices are MySQL, PostgreSQL, and Microsoft SQL Server. Although at this time MySQL is really the only one that is well supported so it is highly recommended that you choose it.
database	host	Please enter your database host or IP address. What is the host that your backend database is running on?
database	user	Please enter your database user name. What is the user name that will be used to access your backend database?
database	passwd	Please enter your database user password. What is the password that will be used to access your backend database?

... continued on next page

Section	Parameter	Description
database	name	Please enter your database name. What is the name of the database that will be used on your RDBMS server?
pyro	hostname	What hostname or IP address should the server bind to? What hostname or IP address should the server bind to. If you leave this empty we will try to guess what host to bind to but be warned that if your hostname resolves to localhost or 127.0.0.1 that you will not be able to connect to the server from any client other that the server itself. If the server has a static IP address you should probably enter that here. If the server has two network interfaces enter the one that will be used for DCS traffic.
pyro	start_name_server	Do you want to automatically start the Pyro Name Server? Automatically starting the Pyro Name Server is the simplest option but there are certain circumstances where you may want to run the Name Server and Event Server separately from the SaberNet DCS Server. For instance if you have other applications on your network that use Pyro.
pyro	ns_hostname	What host is the Pyro Name Server running on? What host is the Pyro Name Server running on? Leving this empty means to broadcast for it.
web	http_port	What port should the HTTP server use? DCS Web runs it's own HTTP server by default (although it can be configured to run through Apache for heavy use). What port would you like the HTTP server to listen on?
web	appserver_port	What port should the AppServer use? DCS Web runs another process that does the heavy lifting called the AppServer. It runs on it's own port. What port would you like it to use? Unless you already have another process using the default port you should probably leave it at the default.
web	xmlrpc_port	What port should the XMLRPC server use? DCS can run an XMLRPC web service server so that custom programmed clients can call certain functionality remotely. It runs on it's own port. What port would you like it to use? Unless you already have another process using the default port you should probably leave it at the default.

... continued on next page

Section	Parameter	Description
operation	description_format	What format should the operation descriptions be in? The operation descriptions are configurable using a format string and substitution variables. The variables are of the form %(variable_name)s where %()s indicates it is a string variable and the variable name is inside of the parenthesis. The available variables include the "Customer Name", "Item Description", "Assembly Description", and "Operation Description". Please see the docs for the full list of available variables. Any characters other that the %()s substitution variables are used verbatim.
pyro	namespace	What Pyro namespace should the sndcs server registered under? What Pyro namespace should the sndcs server registered under? This should probably only need changed if you are running several separate DCS servers on the same network segment or doing some testing. If you are not sure what to do just leave it at the default.

Client Configuration

Section	Parameter	Description
pyro	ns_hostname	What host is the Pyro Name Server running on (empty string means use broadcast only)? The client needs to communicate with the server through the Pyro name server. Usually it can just broadcast for it to find it. In that case you can leave this blank. Under certain circumstances where broadcast doesn't work very well (i.e. vpn connection) or if you get "Name server not found" errors you may need to explicitly set host name or IP address here.
gtk	terminal_id	What terminal ID would you like to give to the GTK+ frontend? The terminal ID is an optional identifier that is recorded with each activity performed by the client. It is mostly useful for reporting purposes if you would like to record the ID of the terminal where clock-in/clock-out/start/stop/etc... were performed.
gtk	fullscreen	Do you want to run fullscreen? Do you want the client to run in fullscreen mode? This is most useful for dedicated terminals where you would not be switching between programs.
gtk	osd_keyboard	Do you want the client to have the OSD keyboard? Do you want the client to have an on-screen keyboard? This is most useful for touch screen terminals.

... continued on next page

Section	Parameter	Description
gtk	search_dialog	Do you want the client to have the search dialog? Do you want the client to a dialog box used for filtering records? This is a useful feature for installations that use a keyboard but if you are using barcodes for all input then this will just take up useless screen real estate.
gtk	menu_bar	Do you want the client to show the menu bar? Do you want the client to show the menu bar? This may be undesirable for a full screen dedicated terminal since the user can exit the application from the menu bar.
gtk	show_clocked_out_employees	Do you want to show clocked out employees in the employee list? Do you want to show clocked out employees in the employee list? If you answer yes you we see clocked out employee in the employee list although they will be greyed out to indicate their clocked-out status. If you answer no then the employee list will only show employees that are currently clocked in.
notes	enabled	Do you want to be able to enter LaborDtl notes? Do you want to be able to enter LaborDtl notes? If you answer yes there will be a field on the end activity screen where miscellaneous notes can be entered.
notes	width	How many pixels wide should the notes text view be?
notes	height	How many pixels high should the notes text view be?
gtk	scrap_codes	Do you want to use scrap codes? Do you want to use scrap codes? Scrap codes allow you create separate categories for reporting scrap quantities. By default there is just a single field for recording scrap quantities. If you answer yes then you can create unlimited types of scrap and each one will allow you to enter a separate scrap quantity when you are ending an activity.
gtk	department_jobs	Do you want to only show jobs where the employee is a member of that department? The available activity list can be filtered to only show jobs belonging to departments that the employee is a member of. If you answer yes the list will be filtered to only show the employee's department jobs. Answering no will make all jobs available.
gtk	qty_complete_warning	Do you want to show a warning if quantity completed would be more than quantity required? When entering the quantity complete on the end activity screen the user can be warned if the total quantity completed would be more than the required quantity.

... continued on next page

Section	Parameter	Description
gtk	recent_activity	Do you want to display recent job activities? The client can display recent jobs to make it easier to clock onto recent activities. If you are using barcodes for all input then this is unnecessary and you should probably enter no.
gtk	easy_employee_selection	Do you want to allow easy employee selection? When easy employee selection is turned on you can select employees from the employee list by double clicking on them with the mouse. If you are doing all input through barcodes you probably want to say no to this.
gtk	force_end_activity	Do you want to force users to fill out the end activity screen? When an employee switches activities it can either be a quick automatic switch or you can force the user to have to see the end activity screen where they can enter quantities, scrap, and/or notes.
gtk	clock_format	What format should the statusbar clock be displayed in? The client's statusbar can display the time on the server for reference. The format should conform to the same format as the python time.strftime() function. An empty string means no time.
pyro	namespace	What Pyro namespace is the sndcs server registered under? What Pyro namespace is the sndcs server registered under? This should probably only be changed if you have set up a DCS server to use a namespace other than the default. If you are not sure what to do just leave it at the default.

Databases

Instructions on creating a DCS database:

- **WARNING: These following steps will delete the database defined if it already exists!!!**

1. "./configure_sndcs.py". (Please read the Installation and Configuration sections before proceeding with this step)
2. When you see the prompt: **"Select your RDBMS Engine"**, pick the one that is applicable.
3. Then enter your database hostname in the following prompt *(i.e 192.168.0.123 or sqlserver.domain)*.
4. And then the username and password that DCS will use, *NOTE: ideally a user that only has permission to the DCS database.*
5. Database name that you want to use, usually best to stick with **'sndcs2'**.
6. You will then see a prompt: **'Do you want to create the database?'** select 'yes', and then follow the instructions of the following prompts.

Base Data

The default user account for DCS Web is: user: administrator passwd: admin

Upgrading

If you are upgrading from a previous installation it is safe to install over top of the previous version. There is no need to explicitly uninstall the previous version. Of course, you will want to make sure that you answer 'No' when prompted to create the database when running the configure_sndcs.py script.

Please check the Release Notes for any important instructions.

Uninstalling

To uninstall SaberNet DCS do the following:

1. Change into the source directory.
2. python setup.py uninstall (as root)

Hosted by SourceForge