|  |
| --- |
| St. Mary's University of San Antonio |
| An Expert System for Network Router Configuration |
| Development Environment Setup Guide |
|  |
| **Michael A. Perez** |
| **4/1/2013** |

|  |
| --- |
| This document is a guide to setup the development environment on Windows and Debian Linux. |

[1.Introduction 2](#__RefHeading__187_1943514952)

[Purpose 2](#__RefHeading__189_1943514952)

[Quick Install Guide: Windows 2](#__RefHeading__191_1943514952)

[Quick Install Guide: Linux 3](#__RefHeading__193_1943514952)

[2.CLIPS Interactive Shell 3](#__RefHeading__195_1943514952)

[3.Interface between Web Server and the CLIPS shell 4](#__RefHeading__213_1943514952)

[4.Key Classes 4](#__RefHeading__223_1943514952)

[Class001: Interview 4](#__RefHeading__225_1943514952)

[Class002: InterviewStream 4](#__RefHeading__227_1943514952)

[UC03: Apply Configuration 4](#__RefHeading__229_1943514952)

[UC04: Save Configuration 5](#__RefHeading__231_1943514952)

[UC05: Load Configuration 5](#__RefHeading__233_1943514952)

[UC06: Load User-defined Knowledge-base 6](#__RefHeading__235_1943514952)

[5.Feature Attributes 6](#__RefHeading__237_1943514952)

[Status 6](#__RefHeading__239_1943514952)

[Priority 6](#__RefHeading__241_1943514952)

[Effort 6](#__RefHeading__243_1943514952)

[Risk 6](#__RefHeading__245_1943514952)

[Stability 7](#__RefHeading__247_1943514952)

[Release 7](#__RefHeading__249_1943514952)

[Assigned-To 7](#__RefHeading__251_1943514952)

[Reason 7](#__RefHeading__253_1943514952)

[6.Software Product Features 7](#__RefHeading__255_1943514952)

[SPF01: Prompt user with yes/no question 7](#__RefHeading__257_1943514952)

[SPF02: Prompt user with multiple choice question 8](#__RefHeading__259_1943514952)

[SPF03: Start wizard 8](#__RefHeading__261_1943514952)

[SPF04: Cancel wizard 9](#__RefHeading__263_1943514952)

[SPF05: Determine if enough information from user has been collected 9](#__RefHeading__265_1943514952)

[SPF06: Package configuration as repeatable script 10](#__RefHeading__267_1943514952)

[SPF07: Configuration script view 10](#__RefHeading__269_1943514952)

[SPF08: Load configuration file from disk 11](#__RefHeading__271_1943514952)

[SPF09: Configuration script editor 11](#__RefHeading__273_1943514952)

[SPF10: Save edits to configuration script 11](#__RefHeading__275_1943514952)

[SPF11: Remotely apply configuration script via SSH to the target SOHO router (TSR) that resides in the local LAN segment. 12](#__RefHeading__277_1943514952)

[SPF12: Determine gateway IP address of current LAN segment 12](#__RefHeading__279_1943514952)

[SPF13: Select IP address of target router 13](#__RefHeading__281_1943514952)

[SPF14: SSH login with username/password credentials 13](#__RefHeading__283_1943514952)

[SPF15: SSH login with public/private key pair credentials 14](#__RefHeading__285_1943514952)

[SPF16: Generate Public/Private key pair 14](#__RefHeading__287_1943514952)

[SPF17: Select Knowledge-base to use for wizard 15](#__RefHeading__289_1943514952)

[SPF18: Load default knowledge-base 15](#__RefHeading__291_1943514952)

[7.Other Product Requirements 16](#__RefHeading__293_1943514952)

[Applicable Standards 16](#__RefHeading__295_1943514952)

[Constraints and Dependencies 16](#__RefHeading__297_1943514952)

[Performance Requirements 16](#__RefHeading__299_1943514952)

[Documentation Requirements 16](#__RefHeading__301_1943514952)

[User Manual 16](#__RefHeading__303_1943514952)

[Installation Guide 17](#__RefHeading__305_1943514952)

[Labeling and Packaging 17](#__RefHeading__307_1943514952)

[Licensing Installation 17](#__RefHeading__309_1943514952)

[8.Glossary And Acronyms 17](#__RefHeading__311_1943514952)

# Introduction

## Purpose

This document will guide the developer in setting up a development environment on Windows 7 and on the Crunchbang Debian Linux distribution. Required libraries, SDKs, and other files are listed.

## Quick Install Guide: Windows

In Windows 7 we are using Cygwin to provide compatible Linux command-line tools including the python runtime and Django 1.5.1 web-framework, Sublime Text 2 for a text-editor, and the Chromium web browser. We are running a 64-bit version of Windows 7 but have installed 32-bit version of Cygwin. We assume you have Windows already installed and that you are familiar with Linux administrative commands such as “cd”, “mkdir”, “git”, “ln”, and bash.

***Installing Chromium web browser:*** Chromium is the open-source project down-stream from the Google Chrome web browser. It has all the functionality you would expect to find in Chrome. As of this writing, chromium.org no longer hosts binary installs for its web browser. However, you can find them at https://commondatastorage.googleapis.com/chromium-browser-snapshots/index.html  
From there you can find installation binaries for Linux, Mac, and Windows. Click on the folder named “Win” for Windows. The next screen is a listing of folders named after release numbers. Find and click on the folder with the largest release number for the latest release of the Chromium. This will usually be the folder at the bottom of the list. You can verify the latest release number by viewing the text file “LAST\_CHANGE” in the Win folder. Within each release folder is a file called “mini\_installer.exe”. Download and run this file to install the Windows version of Chromium on your system.

***Installing Cygwin:*** From cygwin.com, download and run setup\_x86.exe. This program can be run again at any time to upgrade, remove, or install packages for the Cygwin environment. Select a mirror to download the packages from and a categorized list of software will be presented to you. Packages marked “skip” will not be installed. Packages with a version number will be installed. In addition to the default packages already marked for installation for you, be sure mark for installation **gcc**, **gcc-core**, **python** 2.7.x, **mysql** 5.5.x, **wget**, and **git**.

***Installing python and django:*** From within the cygwin shell, use wget to download the setuptools 0.9.8 package from the python package index at pypi.python.org. As of this writing the ez\_setup.py script will install the current version setuptools package for your platform.

>wget https://bitucket.org/pysa/setuptools/raw/bootstrap/ez\_setup.py

If you get certificate errors you may have to modify wget like so:

>wget --no-check-certificate https://bitucket.org/pysa/setuptools/raw/bootstrap/ez\_setup.py

Now run the script and install pip and django 1.5.1.

>python ez\_setup.py

>easy\_install pip

>pip install django==1.5.1

Download the pyclips tarball from sourceforge.net at http://sourceforge.net/projects/pyclips/files/pyclips/pyclips-1.0/ and move it into you cygwin environment.

>mv /cygdrive/c/Users/mperez/Downloads/pyclips-1.0.7.348.tar.gz /home/mperez

>cd /home/mperez

>tar xvzf pyclips\*tag.gz

>cd pyclips\*

>python setup.py install

***Setting up git and getting the code:*** At the cygwin shell prompt, change to a directory you want to store the code and clone the code repository from github.

>git clone git://github.com/meekprize/bakshi

Tip: Your can get to directories outside of the normal cygwin environment by using “/cygdrive/c/” for example to refer to the Windows C: drive

It is easy to develop with source files outside of the cygwin environment using softlinks with the ln command. For example

>mkdir /cygdrive/c/gits

>git clone git://github.com/meekprize/bakshi /cygdrive/c/gits

>ln -s bakshi /home/mperez/bakshi

>cd /home/mperez/bakshi/django\_project

>python manage.py validate

>python manage.py syncdb

>python manage.py runserver 8080

Open Chromium and navigate to http://locahost:8080. The django web app should be running in your web browser.

Running interactive prompt is a great way to quickly test your models.

>python manage.py shell

## Quick Install Guide: Linux

pretty much the same except skip the cygwin step.

# CLIPS Interactive Shell

CLIPS is a shell for running expert systems.

# Interface between Web Server and the CLIPS shell

CLIPS runs in a process separate from the web server. PyCLIPS provides python bindings to the CLIPS shell. We can register python callbacks to be used in CLIPS programs. These callbacks can be used to update databases and assert or redact facts in a CLIPS environment.

# Key Classes

Figure 1: Key Classes (Class Diagram)

## Class001: Interview

Encapsulates the process of an interview session between the user and wizard. A router configuration file will be created at the end of the process in a normal flow.

|  |  |
| --- | --- |
| Brief Description | The Wizard takes the interviewee step by step through the interview process to define the user's home network and determine how it should be configured. |
| Known Subclasses | none |
| Abstract, Interface, or Concrete? | Concrete |
| Attributes | started\_at:datetime  configuration\_file:file |
| Methods | create – returns bool indicating success status of initiating the interview session |

## Class002: InterviewStream

Review Configuration

|  |  |
| --- | --- |
| Brief Description | Though the user is not expected to understand any of it, the set of shell script configuration commands to be run on the target router will be presented for the user's review. |
| References | GN04 |
| Preconditions | Wizard has completed to the end OR the user has loaded a previously saved configuration |
| Normal Flow | The system prompts the user with the configuration commands that will be applied to the target SOHO router. The user may review all the commands before applying them against the target SOHO router. |
| Alt. Flow | none |
| Post Conditions | The user may edit the commands, save the configuration to the file system, and apply the configuration to the target SOHO router. |
| Primary Actors | Admin |

## UC03: Apply Configuration

Apply Configuration

|  |  |
| --- | --- |
| Brief Description | Using SSH, remotely apply the configuration commands to the target SOHO router in the same LAN segment. |
| References | GN01 |
| Preconditions | Target router is installed with OpenWRT with SSH server enabled.  System and target router are on the same LAN segment.  User has SSH login credentials to the target router.  A configuration file is already loaded here after the wizard has run or because the user has loaded a previously saved configuration. |
| Normal Flow | The user opts to apply the configuration currently being reviewed. The system prompts the user for the IP address of the target router. By default, the IP address of the network gateway is used. The user accepts the default or enters in a new IP address. The user is then prompted for the SSH username and password of the remote target SOHO router. Alternately, a public/private key pair may be used instead of a username and password credentials. Using SSH, the system remotely logs into the target router and runs the configuration script. |
| Alt. Flow |  |
| Post Conditions | The SSH connection is closed. The target router is configured according to the configuration commands run by the script. The user's network connection may be interrupted. |
| Exception |  |

## UC04: Save Configuration

Save Configuration

|  |  |
| --- | --- |
| Brief Description |  |
| References | GN01 |
| Preconditions |  |
| Normal Flow |  |
| Alt. Flow |  |
| Post Conditions |  |
| Exception |  |

## UC05: Load Configuration

Load Configuration

|  |  |
| --- | --- |
| Brief Description | Load a previously saved configuration. |
| References | GN01, GN03 |
| Preconditions | Wizard is not running |
| Normal Flow | User loads a previously saved configuration file from disk |
| Alt. Flow |  |
| Post Conditions |  |
| Exception |  |

## UC06: Load User-defined Knowledge-base

Load User-defined Knowledge-base

|  |  |
| --- | --- |
| Brief Description | This project will produce a single knowledge-base flat file to be used in the system. The system will accept user-defined knowledge bases that are compatible with the CLIPS inference engine. |
| References | GN02 |
| Preconditions | none |
| Normal Flow | The user starts the system. From a menu of options, the user chooses a user-defined knowledge base. If the user does not choose a knowledge base, then the knowledge base produced from this project is used. |
| Alt. Flow |  |
| Post Conditions | If the user chooses to run the wizard, the user-defined knowledge base will be used instead of the default knowledge base. |
| Exception |  |

# Feature Attributes

These attributes characterize all product features. Their values should be adjusted to reflect their current state as the project progresses.

## Status

is one of: Proposed, Rejected, Adopted, Implemented

The Status attribute tracks progress during definition of the project baseline and subsequent development.

## Priority

is one of: Critical, Useful, Enhancement

The Priority attribute ranks features by relative benefit to the end user and satisfaction of business goals and needs.

## Effort

is one of: Low, Medium, High

The Effort attribute estimates the amount of time, lines of code, function points, or just general level of effort.

## Risk

is one of: Low, Medium, High

The Risk attribute measures the probability that a feature will cause undesirable events such as cost overruns, schedule delays, or even cancellation.

## Stability

is one of: Low, Medium, High

The Stability attribute measures the level of understanding of a feature.

## Release

is one of: Proposal, Plan, Module1, Module2, Module3, Module4, Module5, …, Final

The Release attribute indicates the intended product version in which the feature will be introduced.

## Assigned-To

The Assigned-To attribute indicates the role or team that is responsible for further elicitation, software requirements, or implementation. Unless otherwise noted, the value for this attribute will be the author.

## Reason

The Reason attribute tracks the source of the requested feature, e.g., one or more goals and needs (NG01-NG05) from section 2 or a Use Case (UC01 - UC-5) from section 4.

# Software Product Features

The following software product features support the realization of one or more of the previously defined use cases.

## SPF01: Prompt user with yes/no question

Prompt user with yes/no question

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC01 |  |

## SPF02: Prompt user with multiple choice question

Prompt user with multiple choice question

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC01 |  |

## SPF03: Start wizard

Start wizard

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC01 |  |

## SPF04: Cancel wizard

Cancel wizard

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC01 |  |

## SPF05: Determine if enough information from user has been collected

Determine if enough information from user has been collected

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC01 |  |

## SPF06: Package configuration as repeatable script

Package configuration as repeatable script

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC01 |  |

## SPF07: Configuration script view

Configuration script view

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC02 |  |

## SPF08: Load configuration file from disk

Load configuration file from disk

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC02, UC05 |  |

## SPF09: Configuration script editor

Configuration script editor.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability |  |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC02 |  |

## SPF10: Save edits to configuration script

Save edits to configuration script.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk |  |  |
| Stability | LOW |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC02 |  |

## SPF11: Remotely apply configuration script via SSH to the target SOHO router (TSR) that resides in the local LAN segment.

Remotely apply configuration script via SSH.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk | LOW |  |
| Stability | HIGH |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC03 |  |

## SPF12: Determine gateway IP address of current LAN segment

Determine gateway IP address of current LAN segment.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk | LOW |  |
| Stability | LOW |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC03 |  |

## SPF13: Select IP address of target router

Select IP address of target router.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk | LOW |  |
| Stability | HIGH |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC03 |  |

## SPF14: SSH login with username/password credentials

SSH login with username/password credentials.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk | LOW |  |
| Stability | HIGH |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC03 |  |

## SPF15: SSH login with public/private key pair credentials

SSH login with public/private key pair credentials.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk | LOW |  |
| Stability | HIGH |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC03 |  |

## SPF16: Generate Public/Private key pair

Generate Public/Private key pair.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk | LOW |  |
| Stability | HIGH |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC05 |  |

## SPF17: Select Knowledge-base to use for wizard

Select Knowledge-base to use for wizard.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk | LOW |  |
| Stability | HIGH |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC06 |  |

## SPF18: Load default knowledge-base

Load default knowledge-base.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Notes** |
| Status |  |  |
| Priority |  |  |
| Effort |  |  |
| Risk | LOW |  |
| Stability | HIGH |  |
| Release |  |  |
| Assigned-To |  |  |
| Reason | UC06 |  |

# Other Product Requirements

Non-functional requirements and their priorities are described here at a high-level. Applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements. The quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set are defined here. If useful, attributes such as priority, stability, effort, and risk are described.

## Applicable Standards

Whenever possible, encryption will be used when sending or receiving configuration data from the network. Consideration for a high Shannon Entropy will guide selection for keys, passwords, and/or algorithms. Follow procedures to ensure configurations are applied only by authorized users.

## Constraints and Dependencies

The system requires TCP/IP network compatible devices.

There is some method available to remotely configure the router.

The system is cross-platform, able to run on Microsoft Windows, Mac OS X, and Linux with X Windows.

## Performance Requirements

The system shall have a peak load of 1 active users.

The system shall have a maximum response time of 60 seconds for target router ping query before timing out.

## Documentation Requirements

Define any specific documentation requirements, including user manuals, online help, installation, labeling, and packaging requirements.

### User Manual

As defined in section 1, a key feature of this system is the accessibility to novice users. Therefore, a user manual will be provided. The Normal and Alternative flows from the Use Cases will guide the creation of the user manual.

### Installation Guide

A guide will be written in which the installation will outlined.

## Labeling and Packaging

The system is developed and packaged as a single unit. Any ancillary systems (i.e. database, server software, etc.) are not provided.

## Licensing Installation

N/A

# Glossary And Acronyms

* AS - see Autonomous System
* Autonomous System - Originally, the definition required control by a single entity, typically an Internet service provider or a very large organization with independent connections to multiple networks, that adhere to a single and clearly defined routing policy, as originally defined in RFC 1771.The newer definition in RFC 1930 came into use because multiple organizations can run BGP using private AS numbers to an ISP that connects all those organizations to the Internet. Even though there are multiple Autonomous Systems supported by the ISP, the Internet only sees the routing policy of the ISP. That ISP must have an officially registered Autonomous System Number (ASN).
* BGP or BGP4 - Border Gateway Protocol (version 4 is the current version)
* Border Gateway Protocol - Core routing protocol of the Internet. It maintains a table of IP networks or 'prefixes' which designate network reachability among autonomous systems (AS). It is described as a path vector protocol. BGP does not use traditional Interior Gateway Protocol (IGP) metrics, but makes routing decisions based on path, network policies and/or rulesets.
* DHCP - Dynamic Host Control Protocol
* Gateway -
* IP - Internet Protocol (version 4 is common, version 6 is reluctantly being adopted)
* ISP - Internet Service Provider
* LAN - Local Area Network
* Local Area Network - The defining characteristics of LANs, in contrast to wide-area networks (WANs), include their usually higher data-transfer rates, smaller geographic place, and lack of a need for leased telecommunication lines. ARCNET, Token Ring, and many other technologies have been used in the past, and G.hn may be used in the future, but Ethernet over twisted pair cabling, and Wi-Fi are the two most common technologies currently in use.
* Router -
* Shannon Entropy - A quantification of the information contained in a message usually in units such as bits. Equivalently, the Shannon entropy is a measure of the average information content one is missing when one does not know the value of the random variable. Shannon's entropy represents an absolute limit on the best possible lossless compression of any communication, under certain constraints: treating messages to be encoded as a sequence of independent and identically-distributed random variables, Shannon's source coding theorem shows that, in the limit, the average length of the shortest possible representation to encode the messages in a given alphabet is their entropy divided by the logarithm of the number of symbols in the target alphabet.
* SOHO - Small Office/Small Home. Consumer-grade. Not enterprise level.
* TCP - Transmission Control Protocol
* TCP/IP - Suite of protocols used in computer IP networks
* VPN - Virtual Private Network
* WAN - Wide Area Network
* Wide Area Network - WANs, in contrast with personal area networks (PANs), local area networks(LANs),campus area networks(CANs), or metropolitan area networks (MANs) are not limited to a room, building, campus or specific metropolitan area (e.g., a city) respectively. The largest and most well-known example of a WAN is the Internet. WANs are used to connect LANs and other types of networks together, so that users and computers in one location can communicate with users and computers in other locations. Many WANs are built for one particular organization and are private. Others, built by Internet service providers, provide connections from an organization's LAN to the Internet. WANs are often built using leased lines. At each end of the leased line, a router connects to the LAN on one side and a hub within the WAN on the other.