

IPautomata Technical Overview

 PROPRIETARY

This document contains material that is the proprietary property of IPsoft. Disclosure outside IPsoft is prohibited except by licensed or other confidentiality agreement.

Copyright © 2007 IPsoft Inc. All rights reserved. Trademarks and service marks of IPsoft include, but are not limited to, eServiceAdvantage, IPmon, IPpm and IPsecure.

TABLE OF CONTENTS

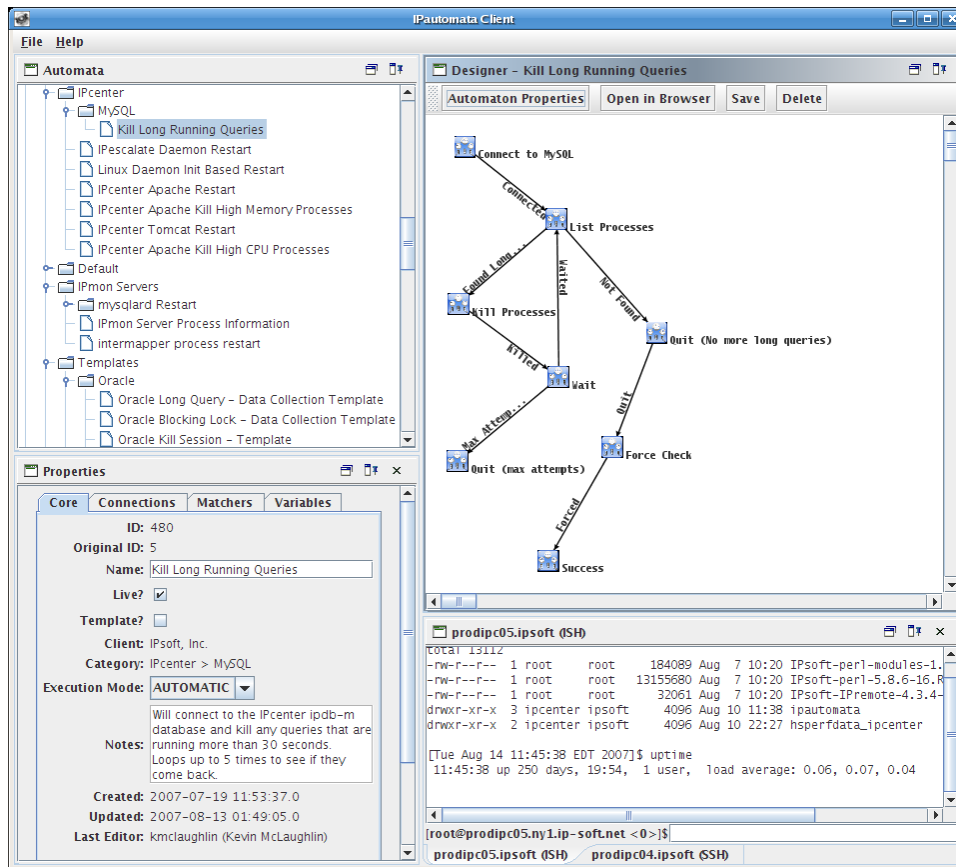
1	About IPautomata	3
1.1	Features	4
1.2	High-Level Architecture	4
1.3	Logging.....	6

1 About IPautomata

IPautomata¹ is a software module that “learns” a remediation sequence from an engineer. Once “taught” how, IPautomata can execute the remediation sequence in response to an incoming IPradar event. (An incoming event may be an alert from IPmon or a ticket from IPpm.)

Execution can be automatic or manually triggered by an engineer. Additional parameters can be supplied by the engineer at runtime for non-automatic automata.

Automata are created in the IPautomata Designer.



IPautomata Designer

The Designer provides a graphical environment where an engineer can spawn shells on remote hosts and execute commands. As the engineer resolves the problem (or event),

¹ *Automaton* (plural, *automata*), *n.* [Etymology: Latin, from Greek, neuter of *automatos*]
 1: a mechanism that is relatively self-operating; especially: robot
 2: a machine or control mechanism designed to follow automatically a predetermined sequence of operations or respond to encoded instructions

the Designer records the engineer's commands and builds a state space, along with transitions based on return codes and/or the output of the commands.

After the initial recording is complete, the Designer provides full editing capabilities allowing the engineer to add/remove states, transitions, create variables, etc.

The IPautomata web GUI provides the ability to view and execute automata as well as the ability to view executions live and historically.

1.1 Features

IPautomata provides the following high-level features:

- Automata creation by both bottom up (recording) and top down learning (manual creation), or a combination of both.
- Multiple connection methods including IPremote, SSH, SSH tunneling, user switching (su, etc.)
- Support for interactive commands (sqlplus, mysql, etc.) that require interactive input from stdin. Requires SSH connection. ("expect" type support).
- Variables – constant variables, secure variables (encrypted), prompt variables, and extract variables via regular expressions or scripting.
- Encrypted communication.
- Full logging of commands executed and their output.
- Linking/sub-automata with full variable scoping (in, out, in_out).
- Templates.
- No additional software required on the remote hosts.
- Browseable history of past executions including full output, commands, timing, etc.
- Utilization of a designated account specifically provisioned for automation.

1.2 High-Level Architecture

IPautomata is composed of four major components:

- **IPautomata Designer:** A graphical design environment where automata are designed. Communicates with remote hosts over supported connection methods and with the IPautomata remote service over HTTPS.
- **IPautomata Remote Service:** Provides remote services for the designer including persistence of automata.
- **IPautomata Execution Daemon(s):** Standalone process that executes automata asynchronously based on a request from the web GUI or automatically based on an appropriate event.

- **IPautomata Web GUI:** Provides HTTPS based front-end to IPautomata within IPcenter. Allows executing automata and viewing the results. The designer is launched via Java web start from the web GUI.

Execution Search											
Status	Creator	Client	Created After	Created Before	Radar Ticket #	IPpm Ticket #	Success	Refresh Every	Pause Refresh	Page Size	Search
READY RUNNING COMPLETE ABORT_REQUESTED ABORTED FAILED	All	IPsoft, Inc.					Yes	1 minute	<input type="checkbox"/>	20	
Generated: 11:54:52									Execution Results		1 - 20 of 655 [Next]
Success	Automaton	Client	Radar Ticket	IPpm Ticket	Status	Created	Finished	Creator			
Yes	IPmon Server Process Information	IPsoft, Inc.	2094770		COMPLETE	06/14 06:22	06/14 06:22				
Yes	IPcenter Apache Kill High CPU Processes	IPsoft, Inc.	2091924		COMPLETE	06/13 13:56	06/13 13:56				
Yes	IPcenter Apache Kill High CPU Processes	IPsoft, Inc.	2091422		COMPLETE	06/13 11:23	06/13 11:24				
Yes	Kill Long Running Queries	IPsoft, Inc.	2091206		COMPLETE	06/13 10:46	06/13 10:47				
Yes	IPcenter Apache Kill High CPU Processes	IPsoft, Inc.	2091196		COMPLETE	06/13 10:43	06/13 10:44				
Yes	IPmon Server Process Information	IPsoft, Inc.	2090627		COMPLETE	06/13 06:46	06/13 06:46				
Yes	IPmon Server Process Information	IPsoft, Inc.	2090622		COMPLETE	06/13 06:43	06/13 06:43				
Yes	Intermapper process restart	IPsoft, Inc.	2090152	369069 / IPsoft-Ops	COMPLETE	06/13 04:03	06/13 04:04				
Yes	Kill Long Running Queries	IPsoft, Inc.	2089961		COMPLETE	06/13 02:00	06/13 02:00	kmclaughlin (Kevin McLaughlin)			

IPautomata Execution Monitoring

IPautomata

Available Automata

[+] IPcenter Apache Restart Edit

[-] IPcenter Apache Kill High CPU Processes Edit

Automaton Notes

Kills IPcenter Apache processes using high CPU. Use this for httpd CPU alerts.

Initial State Name	Command	Host
List Processes	ps auxwww sort -nr -k3 grep '/apps/apache/bin/httpd' grep -v grep	\${host_override}

Name	Value	Notes
host_override	prod1pc03.ipsoft	

[Create New Automaton](#)

Matching Automata Classes/Templates

[Kill High CPU Processes](#) Edit Create Automaton from Template

Automata Executions for Ticket

[-] IPcenter Apache Kill High CPU Processes 06/13 11:23 Success

Execution Complete

Execution History View as Text

[+][-]	Name	Command	Host
[+]	List Processes	ps auxwww sort -nr -k3 grep '/apps/apache/bin/httpd' grep -v grep	prod1pc03.ipsoft
[+]	Store High CPU Processes	ps auxwww sort -nr -k3 grep '/apps/apache/bin/httpd' grep -v grep awk '{if (\$3 > 20) print \$2}' sort -n	prod1pc03.ipsoft
[+]	Wait	sleep 60	prod1pc03.ipsoft
[-]	Check Processes again	ps uwww -p11775 grep '/apps/apache/bin/httpd' grep -v grep awk '{if (\$3 > 20) print \$2}' sort -n	prod1pc03.ipsoft

Run From 11:24:31 to 11:24:31

Return Code: 0

Stdout:
11775

Stderr:

[+] Kill Processes kill -9 11775 prod1pc03.ipsoft

IPautomata Execution Widget and Log

1.3 Logging

All automata executions are fully logged and can be viewed, or exported as text, through the web GUI. The following information is logged for each execution:

- Start and end times of the execution.
- The user who initiated the execution (if manual execution)
- All variable values
- All connection information
- Status, including any reason for failure
- The associated IPradar ticket/event
- The command for each state, including:
 - Start and end time
 - Full command
 - Stdout, Stderr, and return code
 - Remote host
 - Status, including any reason for failure