ZenPack Developers' Guide

Version 1.0.5

DRAFT

Work in progress - for review

Please provide comments to

jane.curry@skills-1st.co.uk

This work is copyright © Zenoss Inc.

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Synopsis

ZenPacks are the extension mechanism provided by Zenoss to build new functionality and also to easily port customisation from one Zenoss server to another. This document is intended to provide information on creating and working with ZenPacks and includes several samples.

The process of creating, modifying and exporting ZenPacks is discussed, along with debugging hints. The sample ZenPacks explore:

- Zenoss architecture
- ZenPack architecture
- Creating new object classes and relationships
- The zenpacklib tool
- Creating new collector modeler plugins to populate the new classes with data
- Converting old, non-zenpacklib ZenPacks to use zenpacklib
- Creating code for web pages for new types of objects, both JavaScript and TAL
- Creating new performance datasources and data templates
- Converting COMMAND-based ZenPacks to use the PythonCollector
- Incorporating new event classes, triggers and notifications in ZenPacks
- Creating and extending menus
- Extending functionality using routers and facades
- Logging and debugging
- The process of ZenPack creation, GitHub and ZenPack submission to Zenoss

✓ Another objective of the paper is to provide examples of good practice. These tips are highlighted throughout the document with a green tick symbol.

The final main objective is to offer deeper insights into the architecture and functionality provided in the standard Zenoss code, especially with reference to how the ZenPack developer uses and extends this code. These sections should probably be skipped initially by someone first starting out with ZenPacks. There are a number of complete sections and they are prefaced with * . Smaller, in-depth points throughout the paper are highlighted with a yellow asterisk.

At the time of writing (Spring 2016) Zenoss 4.2.5 is the platform mainly used in production. Zenoss 5.x is emerging as the platform for new installs and is being slowly adopted by users running earlier versions. Many Zenoss 3 implementations are still in use. This document will major on 4.2.5 practices but will also cover differences for Zenoss 3 and Zenoss 5.

It is assumed that the reader is familiar with basic SNMP and Linux concepts and with standard Zenoss configuration techniques.

This paper was written based largely on Zenoss 4.2.5 with zenup fix 457, on a CentOS 6.3 platform. The hostname of the Zenoss 4 server is *zen42.class.example.org*. Some examples are shown specifically for Zenoss 5 (5.0.7), where the Zenoss 5 server is *zen50.class.example.org*.

Notations

Throughout this paper, text to by typed, file names and menu options to be selected, are highlighted by *italics;* important points to take note of are shown in **bold.**

ℹ Points of particular note are highlighted by a blue “i” icon.

✓ Points of good practice are highlighted with a tick icon.

⭐ Subtle or more advanced points are denoted with a star icon.

ZenPack samples

All the ZenPack samples can be found on GitHub at https://github.com/ZenossDevGuide

Table of Contents

1 Zenoss Architecture....13
1.1 Background to Zenoss....13
1.2 Zenoss concepts....13
1.2.1 Devices, components, object classes and device classes....13
1.2.2 Zenoss monitoring functionality....15
1.2.3 Standard conventions for Zenoss code and ZenPacks....17
1.3 Zenoss Daemons....18
1.4 Zenoss 5 docker architecture....20
1.5 Extending Zenoss out-of-the-box functionality....20
2 What are ZenPacks?....21
2.1 Sources for ZenPacks....21
2.1.1 Free ZenPacks developed by Zenoss....21
2.1.2 Community developed ZenPacks....21
2.1.3 Chargeable Zenoss ZenPacks....22
2.1.4 Write your own ZenPack!....22
2.2 ZenPack basics....22
2.3 Existing ZenPack documentation....23
2.3.1 High-level documentation....23
2.3.2 zenpacklib documentation....24
2.3.3 Standard Zenoss documentation....25
2.3.4 Community ZenPack documentation....25
3 The mechanics of building a ZenPack....25
3.1 ZenPack development environment....25
3.1.1 Zenoss 4 and earlier....25
3.1.2 Zenoss 5....26
3.1.2.1 zenoss user....27
3.1.2.2 Common directory between containers and the base host - /z....29
3.1.2.3 Configuring the service for a development minimum....29
3.1.2.4 Useful references for managing a Zenoss 5 environment....31
3.2 ZenPack creation....31
3.2.1 What's in a name?....31
3.2.2 ZenPack directory hierarchy....32
3.2.3 ZenPack creation for Zenoss 4 and earlier....36
3.2.4 Zenoss 5 ZenPack creation....37
3.2.5 ZenPack creation using zenpacklib....37
3.3 Exporting ZenPacks....38
3.4 Installing ZenPacks....38
3.4.1 Installing ZenPacks on Zenoss 4....40
3.4.2 Installing ZenPacks on Zenoss 5....41
3.5 Removing ZenPacks....41
4 Simple ZenPacks....42

4.1 Adding performance templates to a simple ZenPack....42
4.1.1 Adding SNMP performance templates to a ZenPack....43
4.1.2 Adding zencommand performance templates to a ZenPack....44
4.2 Adding SNMP MIBs and event classes to a simple ZenPack....46
4.3 Adding device classes to a simple ZenPack....49
4.4 * Adding services and processes to simple ZenPacks....49
4.4.1 Adding IP services to a ZenPack....49
4.4.2 Adding Windows Services to a ZenPack....52
4.4.3 Adding Processes to a ZenPack....53
5 Understanding core Zenoss objects....55
5.1 Device.py....56
5.1.1 Object attributes....57
5.1.2 Object relationships....59
5.1.3 Object methods....61
5.2 DeviceComponent.py....63
5.3 * Example object class hierarchy for Fan DeviceComponent....65
5.4 * Example component class relationships for IpInterface....72
5.5 zendmd and the ZMI as tools to understand objects....75
5.5.1 The Zope Management Interface (ZMI)....75
5.5.2 zendmd....78
6 Developing complex ZenPacks....82
6.1 Planning considerations....82
6.1.1 Names and naming convention....82
6.1.2 ZenPack prerequisites and other considerations....83
6.2 zenpacklib....84
6.3 Developing Python code....84
6.3.1 pyflakes....85
6.4 Developing GUI code....86
6.5 Useful tricks for ZenPack developers....86
7 Anatomy of a ZenPack....87
7.1 Basic principles....87
7.1.1 Configuration data, modeler plugins and the zenmodeler daemon....87
7.1.2 Performance data and monitoring templates....91
7.2 New objects in ZenPacks....92
7.3 GUI code....94
7.3.1 Page Template files and skins directories in older Zenoss....94
7.3.2 JavaScript code to define GUI elements....95
7.3.3 configure.zcml, infos and interfaces....95
7.4 Other elements of a ZenPack....99
8 zenpacklib UserGroup sample ZenPack....100
8.1 Requirements specification....101
8.1.1 bash commands to access user and group information....101
8.2 ZenPack specification....102
8.3 Installing zenpacklib....103

8.3.1 PyYAML....103
8.3.2 Installing zenpacklib....104
8.4 Creating the ZenPack....105
8.5 zenpack.yaml....107
8.5.1 zProperties....108
8.5.2 Zenoss device classes....108
8.5.3 Object classes....109
8.5.4 Relationships....114
8.6 Deploying and testing the ZenPack....117
8.7 Modeler plugin....119
8.7.1 Design details....119
8.7.2 UserGroupMap modeler plugin code....120
8.7.2.1 Creating the directory hierarchy....120
8.7.2.2 Imports from other Python modules....120
8.7.2.3 Base class for the UserGroupMap modeler plugin....121
8.7.2.4 Using zProperties in the modeler plugin....122
8.7.2.5 CommandPlugin command....123
8.7.2.6 The process method of the modeler plugin....124
8.7.3 Testing the modeler....132
8.7.4 Where do things go wrong with modelers?....133
8.8 * Renderers....135
8.9 Templates and zenpacklib....137
8.9.1 Creating a User component template with the GUI....138
8.9.2 Exporting templates with zenpacklib....140
8.10 * Creating object methods with zenpacklib....143
8.10.1 Writing methods for objects....144
8.11 *Creating new components directly on Device object class....147
8.11.1 * zenpack.yaml modifications....147
8.11.2 * Other modifications....148
8.11.3 * Testing the changes....149
8.11.4 * Binding device templates in __init__.py....150
8.12 *Creating new components inherited from existing components....153
8.12.1 zenpack.yaml modifications....154
8.12.2 Other modifications....154
8.12.3 Testing the changes....156
9 SNMP LogMatch sample ZenPack....157
9.1 Requirements specification....157
9.2 ZenPack specification....163
9.3 Creating the ZenPack....164
9.4 Creating device and component object classes....165
9.4.1 Checking the device attributes and relationship....167
9.5 Creating the component modeler....168
9.5.1 * SNMP modeler code in core Zenoss....169
9.5.2 The LogMatchMap modeler plugin for component data....171

9.5.3 Testing the modeler....177
9.5.4 The LogMatchDeviceMap modeler for the device....179
9.5.5 Where do things go wrong with SNMP modelers?....180
9.6 GUI display code....181
9.6.1 JavaScript for new components....181
9.6.2 info.py....185
9.6.3 interfaces.py....187
9.6.4 configure.zcml....188
9.6.5 Where do things go wrong with GUI display code?....190
9.6.6 * Architecture of the ComponentPanel....191
9.7 Adding component performance templates....192
9.8 Adding other ZenPack elements through the GUI....194
9.9 Finalising the ZenPack....195
9.10 Extending the ZenPack to modify the device Overview....195
9.10.1 custom-overview-device.js....196
9.10.2 browser/configure.zcml....197
9.10.3 info.py....197
9.10.4 interfaces.py....198
9.10.5 Top-level configure.zcml....198
9.10.6 Testing the new changes....198
9.11 Modifying the ZenPack to remove LogMatchDevice....199
9.11.1 monkeypatching standard objects in __init__.py....200
9.11.2 LogMatch.py....202
9.11.3 browser/configure.zcml....203
9.11.4 LogMatchMap modeler plugin....203
9.11.5 Remove / install ZenPack....204
10 Rewriting the LogMatch ZenPack with zenpacklib....206
10.1 Creating ZenPacks with zenpacklib....206
10.2 zenpacklib capabilities....207
10.3 Converting the logmatch ZenPack for zenpacklib....207
10.3.1 zenpacklib benefits - items no longer required....207
10.3.2 zenpack.yaml....208
10.3.3 zenpack.yaml elements in modeler plugins....211
10.3.4 Completing the ZenPack....211
10.3.5 JavaScript to modify the device Overview panel....212
11 COMMAND DirFile sample ZenPack....213
11.1 Requirements specification....214
11.2 ZenPack specification....214
11.3 Creating the ZenPack....215
11.4 zenpack.yaml....216
11.5 DirFileMap modeler plugin....220
11.5.1 * CommandPlugin code in core Zenoss....220
11.5.2 Using zProperties in the modeler plugin....223
11.5.3 CommandPlugin command....224

11.5.4 The process method of the modeler plugin....226
11.5.5 * What's in an object map?....230
11.5.6 zenpacklib and the modeler plugin....232
11.5.7 Testing the DirFileMap modeler....232
11.5.7.1 * Analysing the zenmodeler log....234
11.6 *monkeypatching so command modeler uses zProperties....237
11.6.1 * Modifying __init__.py....237
11.6.2 * Modifying the modeler plugin code....239
11.6.3 * Testing the new code....243
11.6.4 * Caveats....243
12 Collecting performance data....243
12.1 Testing environment for the ZenPack....243
12.2 Collecting device performance data....244
12.2.1 * Analysing the zencommand debug log....248
12.3 Collecting component performance data....249
12.3.1 Specific component command; single value returned....249
12.3.2 Specific component command; multiple values returned....251
12.3.3 Generic component command with parser....254
12.3.4 Customised datasource to pass customised key values....259
12.3.4.1 getDescription method....263
12.3.4.2 useZenCommand method....263
12.3.4.3 getCommand method....264
12.3.4.4 addDataPoints method....264
12.3.4.5 Infos, Interfaces and configure.zcml....264
12.3.4.6 Testing the new datasource....267
12.4 Performance templates and zenpacklib....269
12.4.1 Where do things go wrong?....271
12.4.1.1 Issues with custom datasources and templates in zenpack.yaml....271
13 Converting COMMAND ZenPacks to PythonCollector....272
13.1 ZenPacks.zenoss.PythonCollector....273
13.1.1 Using the PythonCollector ZenPack....274
13.1.2 * Anatomy of a PythonDataSourcePlugin....276
13.2 Twisted....278
13.3 Creating Python datasources....280
13.3.1 Collecting device performance data....282
13.3.1.1 Imports for the PythonDataSourcePlugin....282
13.3.1.2 proxy_attributes and config_key method for the PythonDataSourcePlugin....283
13.3.1.3 collect method for the PythonDataSourcePlugin....285
13.3.1.4 onResult method for the PythonDataSourcePlugin....289
13.3.1.5 onSuccess method for the PythonDataSourcePlugin....290
13.3.1.6 onError method for the PythonDataSourcePlugin....290
13.3.1.7 Testing the new PythonDataSourcePlugin....291
13.3.1.8 Performance template to drive the PythonDataSourcePlugin....291

13.3.2 * Blocking and non-blocking in Twisted..293
13.3.2.1 * Comparing blocking and non-blocking collect methods..................296
13.3.3 Collecting component performance data; specific component command; single value returned..297
13.3.3.1 Using a dsplugins directory..298
13.3.3.2 Imports, proxy_attributes, config_key and params...........................298
13.3.3.3 * A closer look at the usage of config_keys...299
13.3.3.4 collect method...302
13.3.3.5 onResult, onSuccess and onError methods..303
13.3.3.6 Performance template to drive the PythonDataSourcePlugin...........304
13.3.4 Collecting component performance data; specific component command; multiple values returned. Nagios plugin conversion...305
13.3.4.1 Imports, proxy_attributes, config_key and params...........................306
13.3.4.2 collect method...307
13.3.4.3 onResult, onSuccess and onError methods..307
13.3.4.4 Performance template to drive the PythonDataSourcePlugin...........309
13.3.5 Collecting component performance data; generic component command with parser...310
13.3.5.1 Imports, proxy_attributes, config_key and params...........................311
13.3.5.2 onSuccess method..312
13.3.5.3 Performance template to drive the PythonDataSourcePlugin...........313
13.3.6 Collecting component performance data; customised datasource to pass customised key values...314
13.3.6.1 Building the Python datasource...315
13.3.6.2 Deploying the new datasource...319
13.4 Converting the modeler to use the PythonCollector ZenPack.........................320
13.4.1 Imports...320
13.4.2 Creating a dirRegex directory from zProperties.......................................321
13.4.3 DirFilePythonMap class attributes...322
13.4.4 collect method...323
13.4.5 process method..325
13.4.6 Testing the new modeler..326
13.5 Combining performance data and modeler data...327
14 Events in ZenPacks..330
14.1 Detecting duplicate events..330
14.2 Event auto-clearing mechanism...331
14.3 Exploring the use of event class attributes..331
14.3.1 Detecting “repeat” events...335
14.3.2 Auto-clearing events...335
14.4 Adding transforms to events..337
14.5 Providing event details in a ZenPack..340
14.6 Providing triggers and notifications in a ZenPack...342
14.6.1 * Trigger and notification architecture..343
14.6.1.1 Finding trigger details..343

14.6.1.2 Finding notification details....................346
14.6.1.3 Dumping trigger and notification details....................348
14.6.2 ZenPack file for triggers and notifications....................348
14.7 Resolving issues with triggers and notifications....................349
14.8 Known issues with event fields, notifications and triggers....................350
15 Creating menus in ZenPacks....................351
15.1 The jargon....................351
15.1.1 Zenoss 2 (some of which is still relevant!)....................351
15.1.2 Zenoss 3 / 4 / 5....................353
15.2 Extending Command menus with the GUI....................354
15.3 ZenPacks.community.MenuExamples....................355
15.3.1 New device class, device object class and component class....................357
15.3.2 Menu defined in __init__.py....................357
15.3.3 Old and new options for page templates for menus....................358
15.3.4 New-style menus limited to specific device types....................363
15.3.5 Dropdown menus shipped in objects.xml....................364
15.3.6 Adding items to the Display dropdown for a component....................371
15.3.7 Menu on INFRASTRUCTURE -> Devices to add new device type....................373
15.3.7.1 Routers and facades....................377
15.3.8 New items for left-hand DeviceClass Action menu....................378
15.3.9 Adding new items to a device's Action menu....................381
15.3.10 Adding a new menu to the Footer bar....................384
16 Testing and debugging ZenPacks....................390
16.1 Log files and logging....................390
16.1.1 Log messages and their likely meanings....................391
16.2 General hints and tips....................392
16.3 Testing and debugging new object class files....................393
16.3.1 New components do not appear in left-hand menu....................393
16.4 Testing and debugging modeler plugins....................394
16.4.1 Compilation errors....................394
16.4.2 General modeler debugging hints....................396
16.4.3 Attributes or relationships are not populated....................397
16.4.4 Modeler issues related to using zenpacklib....................399
16.5 Testing and debugging problems with performance data....................400
16.5.1 Configuration issues....................400
16.5.2 Checking for collected performance data....................401
16.5.3 Test buttons in datasources....................403
16.5.4 Issues with datasource plugins....................404
16.5.5 Issues with datasources....................404
16.5.6 Performance collection issues related to using zenpacklib....................405
16.6 Testing skins files and JavaScript files....................405
16.6.1 General failure errors....................406
16.6.2 Problems displaying components....................408
16.6.3 Issues with Info and Interface definitions and configure.zcml....................408

16.6.4 GUI issues when using zenpacklib....409
16.7 Testing and debugging problems with event elements....409
16.8 Problems with installing / removing ZenPacks....410
17 Developing a ZenPack and making it publicly available....411
17.1 Simple procedure for git development....411
17.2 Working with GitHub....414
17.2.1 Using ssh authentication with GitHub....415
17.2.2 Creating the GitHub repository....416
17.3 git branches....417
17.4 Cloning from GitHub to a local machine....417
17.5 Other ways to use GitHub....418
17.6 ZenPacks on the Zenoss wiki....418
References....424
ZenPack Reference....429
Acknowledgements....431
About the author....431