vw4tools

vw4tools

Table of Contents

List of Figures

List of Tables

Chapter 1. Namespace Documentation

com::fasterxml::jackson::core

com::fasterxml::jackson::core::json

com::fasterxml::jackson::databind

com::fasterxml::jackson::databind::jsonschema

org

Namespaces

- struct org::smallfoot

org::smallfoot

Namespaces

- struct org::smallfoot::vw4

org::smallfoot::vw4

Classes

- struct org::smallfoot::vw4::Entity

- struct org::smallfoot::vw4::EntityFA

- struct org::smallfoot::vw4::EntityHBA

- struct org::smallfoot::vw4::VirtualWisdom4ClientTool

Chapter 2. JavaDoc API Markup for vw4tools

vw4tools .

Tools and utilities for VirtualWisdom4

At current, VW4Tools is a script or two, plus one big jar file. vw4tool.jar allows a user to import a text file in one of several formats and output an Entity import JSON for VirtualWisdom4.

Running Java .

In this document, it is assumed "java" runs from the command prompt. In some cases, this isn't true. Windows users, for example, may need to change their usage, from:

```
java -jar vw4tool.jar ...
```

to:

```
java.exe -jar vw4tool.jar ...
```

In both cases, if the "java" (or java.exe) executable is not available on the user's "path" or list of locations wherein to find a binary/executable, the user will need to type the full pathname to the java executable.

Macintosh OSX users can simply type "java":

```
java -jar vw4tool.jar ...
```

A UNIX (or unix-like) user may need to type /opt/local/bin/java or similar:

```
/opt/local/bin/java -jar vw4tool.jar ...
```

A Windows user may need to type a full pathname such as:

```
"C:\Program Files (x86)\Java\jre7\bin\java.exe" -jar vw4tool.jar ...
```

In all cases, where I type merely "java", make sure you replace that with however the invocation details are for your platform.

To confirm you've found a working Java Runtime Environment, use the "help" command to as vw4tool.jar what version it is:

```
java -jar vw4tool.jar --help
Usage: vw4tool -V|--version|-H|--help
     : vw4tool --read <filename>|--input <filename> | -r <filename> | -i <filename
   ie: vw4tool --read import.json
   ie: vw4tool -r import.json
```

note: many commands have shorter versions. "--version" (with two "-") is the same as "-V" (with one dash). Pay special attention to case. Uppercase "N" is different from lowercase "n".

Common Usage .

```
java -jar vw4tool.jar  -N localfile1.txt -N localfile2.txt -N localfile3.txt -oexa
```

or

```
java -jar vw4tool.jar  -N localfile1.txt -N localfile2.txt -N localfile3.txt -oOrd
awk -f csv-to-json.awk OrderedTuples.csv > example.json
```

This is the most common usage, here as a reference. Notice that although this shows EITHER OrderedTuples.csv or a JSON file being created, both could be created by multiple "-o" options. Notice also that "-o" is "cuddled" to the filename after it. No spaces, but this should change in the future to avoid ambiguity.

Overview .

In general, collection and conversion looks like the following diagram:

Figure 2.1.

Import of FTP or HTTP Data .

VW4Tools uses the URLDataSource functionality to open a stream from a HTTP or FTP server; this means that the files it parses can be stored on a local filesystem, a FTP server, and an HTTP server as constant data, or can be the result of a CGI program on an HTTP server. Less common, a local fil emay also be mounted as a FUSE filesystem, allowing even "local" data to be generated like a CGI program.

In order to draw a text stream from a FTP server or HTTP server, merely use a RFC-1738-compliant URL to indicate these two sources:

```
(ftp|http)://{user{:password@@}server{:port}/pathname/to/resource
```

for example:

(basic user/pass: user = scott, pass = T1ger, host = ftp.example.com, subdir = . , file = nicknames.txt

```
java -jar vw4tool.jar --nickname=ftp://scott:T1ger@@ftp.example.com/nicknames.txt
```

(basic user/pass: user = scott, pass = T1ger, host = ftp.example.com, subdir = /users/local/ScottAdams/ , file = aliases.text

```
java -jar vw4tool.jar --nickname=ftp://scott:T1ger@@ftp.example.com/%2fusers/local
```

(basic user/pass: user = anonymous, pass = scott@uberserver.net, host = ftp.example.com, subdir = examples , file = aliases

```
java -jar vw4tool.jar -N ftp://anonymous:scott%2Fuberserver.net@@ftp.example.com/e
```

(basic http: host = www.example.com, subdir = . , file = aliases

```
java -jar vw4tool.jar --nickname=http://www.example.com/aliases
```

(basic HTTP GET: host = www.example.com, subdir = . , path=/cgi-bin/fetch.cgi, some parameters

```
java -jar vw4tool.jar --nickname=http://www.example.com/cgi-bin/fetch.cgi?now=2014
```

NOTE: "--nickname=" and "-N" are functionally identical; "--nickname=" may offer a slight beefit in more clearly self-documenting the behavior of the commandline option.

Import of Local Text File .

The most common usage is local text files representing the metadata of the local environment. This is done by offering a RFC-1738-compliant local file URL:

```
file://pathname/to/resource
```

Note that if the "file://" is not given, and no "protocol" (ie "file://") is given, vw4tool will warn you about this but try the local file:// protocol prefix for you

for example:

(local file in subdir ./files/aliases)

```
java -jar vw4tool.jar --nickname=file://files/aliases
java -jar vw4tool.jar --nickname=files/aliases
```

(local file in subdir /Full/Path/files/aliases)

```
java -jar vw4tool.jar --nickname=file:///Full/Path/files/aliases
java -jar vw4tool.jar --nickname=/Full/Path/files/aliases
```

Specifying Formats .

The user doesn't need to speficy the format of a file; vw4tool will try the file at the same time across many different parsers and see which one can make sense of it. Unfortunately, the parsers cannot always chew through any user-interface codes (such as "press any key to continue") and preamble (the verbose text trash before the actual zone or aliases or such). Trimming that to a minimum offers a better chance of parsing the file.

The corollary to this is that if a file doesn't seem to parse, yet it seems like it should, remove anything before or after the actual content, and confirm that it was collected in a non-interactive method. If the user ever needs to "press any key for more", chances are, the parsed result will be either partial, or none at all.

Multiple Inputs .

These input formats can be mixed. For example

```
java -jar vw4tool.jar --nickname=http://www.example.com/cgi-bin/fetch.cgi?now=2014
```

Outputs .

vw4tool will either generate an OrderedTuples.csv file (if that specific filename is given on the "-o" option) or a json file (any other putput filename). Or both:

```
java -jar vw4tool.jar  -N local.txt -oOrderedTuples.csv -oexample.json
```

NOTE: giving no filename sends the output to stdout:

```
java -jar vw4tool.jar  -N local.txt -o
```

For this reason, the following two commands do different things:

```
java -jar vw4tool.jar  -N local.txt -oexample.json
java -jar vw4tool.jar  -N local.txt -o example.json
```

Stdout can be represented as "-"; the following two commands do the same thing:

```
java -jar vw4tool.jar  -N local.txt -o
java -jar vw4tool.jar  -N local.txt -o-
```

This behavior may disappear in future. Please try to use "-" to print the result to stdout:

```
java -jar vw4tool.jar  -N local.txt -N (another) --nickname=(another) -o-
```

Proprietary Intellectual Property .

With the exception of samples/import01.json, all files and content are based on the same level of access to the product enjoyed by a customer. Implicitly, no private information is shared, all of this content (save the one file) is based on empirical discovery, and could change overnight.

Chapter 3. README

Chapter 4. Todo List

Global VirtualWisdom4ClientTool._load (String filename)	:	evaluate: mapper.configure(DeserializationConfig.Feature.FAIL_ON_UNKNOWN_PROPER false);

Chapter 5. Class Documentation

org::smallfoot::vw4::Entity class Reference

Classes

- struct org::smallfoot::vw4::Entity::ImproperChildException

- struct org::smallfoot::vw4::Entity::LeafEntity

Protected Attributes

- Vector< Entity > children

  local singleton late-instantiated as needed by children()

- String description

  the description of the entity showing source

- String name

  the unique name of the entity

- WeakReference< Entity > parent

  convenience weak-reference to parent: I want a reference but not one that will block GC

- abstract boolean canBeChild (Entity e)

- Vector< Entity > children ( )

- abstract org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

- Entity (String name)

- Entity (String name, Entity e)

- String description ( )

- String name ( )

- void setDescription (String description)

- void setname (String name)

- int addTo (VWImport v, String tag)

- boolean isOrphan ( )

- boolean isOrphan (TreeMap< String, Entity > list)

Brief Description

An Entity is the core mutable object used in the JSON import for VW4.

Detailed Description

I'm not so sure how it'll unfold yet, but I intend to treat all hosts and hbas and storagecontrollers and iomodules as similar objects: things that can have children things. In the short term, be very careful: there is an Entity, and a VWImport::Entity

Definition at line 47 of file Entity.java

The Documentation for this struct was generated from the following file:

- Entity.java

abstract boolean canBeChild (Entity e)

whether a given entity can be this entity's child

Returns: .   true if accepted, false if refused

Table 5.1. Parameters

e	entity to check for possible descendent-hood

Vector<Entity> children ()

abstract org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

create a streamable JSON entity from this one

Returns: .   a org.smallfoot.vw4.VWImport.Entity representation of this instance

Table 5.2. Parameters

tag	default tag to apply

Entity (String name)

Class Constructor with no initial child.

Entity (String name, Entity e)

Class Constructor with an initial child to absorb.

String description ()

String name ()

void setDescription (String description)

void setname (String name)

int addTo (VWImport v, String tag)

convenience: add myself and all children to the streamable exporter given as "v"

Table 5.3. Parameters

v	VWImport streamer to add myself and children to
tag	default tag to apply

Returns: . number of entities written

boolean isOrphan ()

boolean isOrphan (TreeMap< String, Entity > list)

org::smallfoot::vw4::EntityFA class Reference

- EntityFA (String name, String wwn)

- org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

Brief Description

An EntityFA is the representation of an Storage FA entity in the JSON import for VW4.

Detailed Description

Be very careful: there is an Entity, and a VWImport::Entity

Definition at line 45 of file EntityFA.java

The Documentation for this struct was generated from the following file:

- EntityFA.java

EntityFA (String name, String wwn)

Basic Class Constructor.

org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

create a streamable JSON entity from this one

Returns: . a org.smallfoot.vw4.VWImport.Entity representation of this instance

Table 5.4. Parameters

tag	default tag to apply

org::smallfoot::vw4::EntityHBA class Reference

- EntityHBA (String name, String wwn)

- org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

Brief Description

An EntityHBA is the representation of an HBA entity in the JSON import for VW4.

Detailed Description

Be very careful: there is an Entity, and a VWImport::Entity

Definition at line 45 of file EntityHBA.java

The Documentation for this struct was generated from the following file:

- EntityHBA.java

EntityHBA (String name, String wwn)

Basic Class Constructor.

org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

create a streamable JSON entity from this one

Returns: . a org.smallfoot.vw4.VWImport.Entity representation of this instance

Table 5.5. Parameters

tag	default tag to apply

java::lang::Exception class Reference

org::smallfoot::vw4::Entity::ImproperChildException class Reference

- ImproperChildException (String s)

- ImproperChildException (Entity e, Entity p)

Brief Description

Descendents of Entity should know whether a given entity can be one of their child elements.

Detailed Description

This exception is intended as a method of signalling – and rippling up if necessary – than an intended seconding as a child element is not accepted by the would-be parent.

Definition at line 50 of file Entity.java

The Documentation for this struct was generated from the following file:

- Entity.java

ImproperChildException (String s)

create a new instance with the given message

Table 5.6. Parameters

s	description to encapsulate

ImproperChildException (Entity e, Entity p)

create a new instance with a consistent message based on the entities offered

Table 5.7. Parameters

e	child entity
p	parent entity

org::smallfoot::vw4::Entity::LeafEntity class Reference

Protected Attributes

- String wwn

  the unique WWPN of the hba

- LeafEntity (String name, String wwn)

- String wwn ( )

- boolean canBeChild (Entity e)

- org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

Brief Description

A LeafEntity is the common ancestor of Storage FAs and Server HBAs; this is combined only so that leaves can be treated in common.

LeafEntity (String name, String wwn)

Class Constructor with no initial child.

String wwn ()

boolean canBeChild (Entity e)

org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

create a bogus function to avoid build errors

org::smallfoot::vw4::VirtualWisdom4ClientTool class Reference

Private Attributes

- org.w3c.dom.Document xmlDocument

eventually used to hold an XML document when converting XML<–>JSON<–>XML

Public Attributes

- VWImport vwimport

singleton list of JSON-writable objects accessed through vwimport()

Protected Attributes

- TreeMap< String, Entity > entities

local singleton array accessed from entities()

- Vector< Pattern > patterns

local singleton array accessed from patterns()

- WWNDescription wwndesc

local singleton array accessed from wwndesc()

- VirtualWisdom4ClientTool (String xmlFile)

- VirtualWisdom4ClientTool ( )

- TreeMap< String, Entity > entities ( )

- void load (String filename)

- Vector< Pattern > patterns ( )

- String prettyJSON (VWImport v)

- void save (String filename)

- WWNDescription wwndesc ( )

- void _load (String filename)

- void _save (String filename)

- void loadAndAbsorbFile (String f)

- void loadAndRemoveFile (String f)

- VWImport vwimport ( )

- void absorbPattern (String pattern)

- Entity entityAccepting (Entity newguy)

- String[] exercisePattern (String alias)

- int finalizeEntities ( )

- void usage (String progname)

- void writeOrderedTuples (String name)

- void writeOutputfile (String filename)

- static void main (String args)

Brief Description

VirtualWisdom4ClientTool is a "Swiss Army Knife" of tools used when working with VirtualWisdom4.

Detailed Description

The existence of these tools is not a judgement on VirtualWisdom4's proper Engineering; rather, an acceptance that a faster-response solution for the longer-tail of the normal curve is often helpful swapping QA delay for reduced customer friction.

As you'd expect, there is no support for this. If it breaks, you may choose to keep both pieces :)

Ad-Hoc content for this utility-stack may appear at http://fcfae.com/

Definition at line 57 of file VirtualWisdom4ClientTool.java

The Documentation for this struct was generated from the following file:

- VirtualWisdom4ClientTool.java

VirtualWisdom4ClientTool (String xmlFile)

Class Constructor to create with an initial file to load.

Table 5.8. Parameters

xmlFile	File to load at start

See also: .   load(String)

VirtualWisdom4ClientTool ()

Class Constructor with no initial file.

TreeMap<String,Entity> entities ()

void load (String filename)

Wrapper to just load the file, spitting out exceptions and stacks as they occur.

Table 5.9. Parameters

filename	file to load

Vector<Pattern> patterns ()

String prettyJSON (VWImport v)

Convenience function to generate a pretty-printed JSON text string.

Table 5.10. Parameters

v	VWImport object to markup

Returns: .   a pretty-printed JSON using ObjectWriter.withDefaultPrettyPrinter() or null if an exception occurs

void save (String filename)

Wrapper to just save the file, spitting out exceptions and stacks as they occur.

Table 5.11. Parameters

filename	filename to save into

WWNDescription wwndesc ()

void _load (String filename)

Open a file.

This is actually a wrapper for the underlying file load

Table 5.12. Parameters

filename	file to load

Table 5.13. Exceptions

IOException	when File() encounters an error instaitating (typically path or permissions)

void _save (String filename)

Save the current XML Document to a new file.

Table 5.14. Parameters

filename	filename to save into

void loadAndAbsorbFile (String f)

one-shot load a new file and absorb the contents: open the file and stream the contents at an array of parsers, the one with the best results wins; using that result, absorb all alias information, attempting to create parent storagecontroller(s) or hosts as resulting from absorbtion patterns

Table 5.15. Parameters

f	name of file to open

See also: .   https://github.com/chickenandpork/fibrechannel-parsers/

void loadAndRemoveFile (String f)

one-shot load a new file and remove the contents from the internal list of leafEntities (HBAs, FAs)

Working on only the leaf entities, this uses the parser array to parse an input stream, and for each WWPN found, it removes that leaf entity from the system. It doesn't affect the child_entities of referring entities to allow for a loaded entity list to still refer to entities which are already on the target system.

Table 5.16. Parameters

f	name of file to open

See also: . https://github.com/chickenandpork/fibrechannel-parsers/

VWImport vwimport ()

void absorbPattern (String pattern)

Absorb a pattern for chunking together nickname patterns.

Hosts will absorb new matches at a top-level (host->hba); storage will absorb a pattern by force into a second-tier (storagearray -> storagecontroller). This will let devices with non-intuitive combinations (ie SVC nodes in an iogroup don't have any matching WWPN parts). Not sure how far this will snowball.

param pattern a String pattern to absorb (into an internal array for use later)

Entity entityAccepting (Entity newguy)

String [] exercisePattern (String alias)

Exercise all current chunking patterns against a given nickname.

Given a string assumed to be a hostname or alias, all patterns currently absorbed will be exercised to show the possible results in order they would be attempted. This is intended to allow in-situ pattern editing and verification rather than perfect knowledge of the regex system used.

As is assumed in the VW4 product, the regex is performed as: s/{pattern}/\1/ such that for a given pattern, the first capturing group is the result For example, given a pattern of "^(.*).{5}$", the ".{5}$" matches the final 5 characters (5 wildcards before end-of-line), leaving the remainder within "^(.*)". All but the final 5 characters are contained in that capture-grouping, so the result is all but the final 5 characters.

param alias a String nickname to exercise all patterns against

int finalizeEntities ()

Finalize absorbed Entities: for each entity, transfer from structured tree of Entities to vwimport array for writing.

Returns: . number of items finalized

void usage (String progname)

Usage messages are useful to those of us with short memories as well (hey, I just need to add swap!)

Table 5.17. Parameters

progname	the name of the program (args[0]) to print on a usage message

void writeOrderedTuples (String name)

Export an "OrderedTuples.csv" file as a stopgap measure until proper JSON processing can be completed.

Table 5.18. Parameters

name	filename to write

Table 5.19. Exceptions

java.io.IOException	when PrintStream throws an exception

void writeOutputfile (String filename)

Write an output file, finalizing from entities( ) if required.

Table 5.20. Parameters

filename	name of file to write

static void main (String args[])

Main function, as you can tell.

this function merely parses the command-line to dispatch actions to the meat of the meal above. I'm using an actual GetOpt because, yes, I'm a UNIX/C hack, re-using 3-decades-old knowledge, but this also preserves both extensibility and the ability to use longopts in scripts as a way to self-document what the tool's doing.

No real intelligence herein except the parse-and-redirect action.

Table 5.21. Parameters

args	as you'd expect, these are commandline arguments given when the jar is activated

Chapter 6. File Documentation

README.dox File Reference

Detailed Description

Definition in file /Users/allan.clark/src/vw4tools/htdocs/README.dox

Entity.java File Reference

Classes

- struct org::smallfoot::vw4::Entity
- struct org::smallfoot::vw4::Entity::ImproperChildException
- struct org::smallfoot::vw4::Entity::LeafEntity

Namespaces

- struct com::fasterxml::jackson::core
- struct com::fasterxml::jackson::core::json
- struct com::fasterxml::jackson::databind
- struct com::fasterxml::jackson::databind::jsonschema
- struct org::smallfoot::vw4

Detailed Description

Definition in file /Users/allan.clark/src/vw4tools/java/Entity.java

EntityFA.java File Reference

Classes

- struct org::smallfoot::vw4::EntityFA

Namespaces

- struct org::smallfoot::vw4

Detailed Description

Definition in file /Users/allan.clark/src/vw4tools/java/EntityFA.java

EntityHBA.java File Reference

Classes

- struct org::smallfoot::vw4::EntityHBA

Namespaces

- struct org::smallfoot::vw4

Detailed Description

Definition in file /Users/allan.clark/src/vw4tools/java/EntityHBA.java

VirtualWisdom4ClientTool.java File Reference

Classes

- struct org::smallfoot::vw4::VirtualWisdom4ClientTool

Namespaces

- struct org::smallfoot::vw4

Detailed Description

Definition in file /Users/allan.clark/src/vw4tools/java/VirtualWisdom4ClientTool.java

VWImport.java File Reference

Namespaces

- struct org::smallfoot::vw4

Detailed Description

Definition in file /Users/allan.clark/src/vw4tools/java/VWImport.java

Chapter 7. Directory Documentation

java Directory Reference

Files

- file Entity.java

- file EntityFA.java

- file EntityHBA.java

- file VirtualWisdom4ClientTool.java

- file VWImport.java

Detailed Description

Directory location is /Users/allan.clark/src/vw4tools/java/