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::EntityArray
- struct org::smallfoot::vw4::EntityFA
- struct org::smallfoot::vw4::EntityHBA
- struct org::smallfoot::vw4::EntityHost
- 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 although in past, "-o" was "cuddled" to the filename after it (no spaces), but this has changed to make it easier and more forgiving to use.

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 file may 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 used to send the output to stdout, but now requires a "-" to mean "yes, I meant that":

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

For this reason, the following two commands used to do different things, but are now equivalent:

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

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. Scripting Options

csv-to-json.awk: Convert a "OrderedTuples.csv" file to a JSON Entity import

csv-to-json.awk: Convert a "OrderedTuples.csv" file to a JSON Entity import

Overview

This script (csv-to-json.awk) interprets a basic 4-column CSV into a JSON entities file. This is "the old method", but facilitates hand-tweaking if required. In essence, the four columns of the file are:

1. host/array name
2. device type: "host" or "array" (without quotes)
3. WWPN
4. optional unique name of the WWPN within the (column 1) host

For example:

Table 4.1.

name	type	WWPN	(optional) name
MPV123456-001	host	2001001560123456	MPV123456-001
MPV123456-013	host	2013001560123456	MPV123456-013
NetApp-123456	array	500a098598123456	
NetApp-123456	array	500a098698123456	
Oracle01	host	10:00:00:00:c9:12:34:56	Oracle01-A
Oracle01	host	10:00:00:00:c9:12:34:57	Oracle01-B

Running this Script

This AWK script should work with gawk.exe from the UnxUtils project on sourceforge.net (which offers windows binaries without need of cygwin) and is regression-tested using basic awk on BSD-based operating systems.

```
gawk.exe   -f csv-to-json.awk   OrderedTuples.csv   >   customer-name.json
```

Commandline Variables:

- -v SKIPARRAY={something} will avoid exporting the array which contains iomodules
- -v SKIPHOST={something} will avoid exporting the host which contains hbas

Usage (2 switches)

```
plink.exe -l username -pw pAssw0rd 192.168.1.44 'show device-alias database' > SW4
plink.exe -l username -pw pAssw0rd 192.168.1.42 'zoneshow' > SW42.zoneshow
java.exe   -jar vw4tool.jar   -N SW44.dad   -N SW42.zoneshow   -oOrderedTuples.csv
```

(manual changes to the OrderedTuples.csv file)

```
sort.exe OrderedTuples.csv    | gawk.exe   -f csv-to-json.awk   >  customer-name.j
```

If the user wants to avoid showing hosts in that output (but still produce HBAs):

```
plink.exe -l username -pw pAssw0rd 192.168.1.44 'show device-alias database' > SW4
plink.exe -l username -pw pAssw0rd 192.168.1.42 'zoneshow' > SW42.zoneshow
java.exe   -jar vw4tool.jar   -N SW44.dad   -N SW42.zoneshow   -oOrderedTuples.csv
sort.exe OrderedTuples.csv    | gawk.exe   -v SKIPHOST=gary.yuen   -f csv-to-json.
```

It is assumed the user wants to make changes to OrderedTuples.csv before the final step; otherwise, the last two steps may be combined (which avoids this script):

```
plink.exe -l username -pw pAssw0rd 192.168.1.44 'show device-alias database' > SW4
plink.exe -l username -pw pAssw0rd 192.168.1.42 'zoneshow' > SW42.zoneshow
java.exe   -jar vw4tool.jar   -N SW44.dad   -N SW42.zoneshow   -ocustomer-name.jso
```

Chapter 5. JVM Options

Global Entity._compatibilityVersion ()	**compat.target** (values: X.Y.Z as a version. For example: 4.0.1, 4.1, 4, 4.2) can be used to tell the JSON output to use specific features available on later versions of the VirtualWisdom4 product. Initially controls whether fcport entities are created but may expand.
Global VirtualWisdom4ClientTool.setTarget (String ver)	**debug.showCompatTarget** causes the compat.target to be displayed when (re)set

Chapter 6. Todo List

Global VirtualWisdom4ClientTool._load (String filename)	: evaluate: mapper.configure(DeserializationConfig.Feature.FAIL_ON_UNKNOWN_PROPER false);
Global VirtualWisdom4ClientTool.main (String args[])	need to check 3-decimal values such as "-t 4.0.1.7" (see also testsuite.at.in)

Chapter 7. Commandline Options

Global VirtualWisdom4ClientTool.main (String args[])

-H|–help Show a simple help screen as a reminder of options which are understood by the application

```
java -jar vw4tools.jar --help
```

-V|–version Show the current release version for reference

```
java -jar vw4tools.jar --version
 0.9-122
```

-n|–nicknameout={file} Output nicknames from internal store

-o|–output={file} Output nicknames from internal store – nicknameout and –output are currently functionally identical; they both cause the internal nickname/entity base to be written out as JSON with the exception of a few "magic" filenames:

1. **schema.json** will cause the current schema to be written

2. **orderedtuples.csv** will cause an OrderedTuples.csv file to be written, suitable for post-processing via csv-to-json.awk but allowing a user to more easily edit CSV for fine-tuning

3. **orphanentities.csv** will cause a CSV to be written listing all orphan entities. An "Orphan Entity" is an entity lacking a parent entity, such as an "HBA Port" without a "host" parent, or a "iomodule" without a parent "storagearray" entity.

All other filename patterns will result in a JSON-formatted file

-N|–nickname={file/uri} Import nicknames by parsing a text stream from various sources

```
java -jar vw4tools.jar --nickname=switch44.zoneshow
      Parse results for AliShowZoneParser:
      Zones: 44
      Aliases: 112 (names with one or more WWPNs)
      Aliases: 136 (name/WWPN tuples)
```

In this example, a zone file was parsed by the AliShowZoneParser resulting in 112 nicknames; due to duplicate nicknames, there are actually 136 unique WWPN/alias tuples, which means that (136-112) 24 of the WWPNs have the same alias as other WWPNs

-i|–input import an existing JSON file for later editing

-r|–read import an existing JSON file for later editing

```
java -jar vw4tools.jar --read working.json
```

-R|–remove Parse nicknames for removal from the internal nickname list

-R|–removenicknames Parse nicknames for removal from the internal nickname list

-t|–target set a target version for JSON output. Defaults to latest supported. A value of 4.0.1 is converted to 40.1 internally due to float comparison and expected changed every bump of major/minor version

-!|–report Summarize the current status of the internal nicknaes and pattern/collation coverage

```
java -jar vw4tools.jar --nickname=switch44.zoneshow  --re
(vw4tools) parsed 0 zones, 2 aliases via Alias4Parser
vw4tools 0.9-122
    5 total entities
    4 leaf nodes
    2 orphans
50.00 % coverage
```

-P|–pattern= is used to provide an "aggregating pattern" to collect Orphan Entities into a container. An "Orphan Entity" is an entity which is not part of a larger device: an HBA not assigned to a host, or a FA not assigned to a storage array. Aggregating Patterns are evaluated immediately, so their order amidst other command options to import or remove entities is important.

-p|–checkpattern= is used to test an "aggregating pattern" against a certain nickname/alias. Although Aggregating Patterns are evaluated immediately, they're stored for testing as well, so even though we're giving a test alias after the pattern, it'll evaluate all loaded patterns against the alias:

```
 java -jar vw4tools.jar --pattern="([^-]+)-[^-]+$" --chec
"NetApp-123456" --> "NetApp"
"UberServer_44_HBA0" --> "UberServer_44_HBA0"
```

in this example, the one pattern it tested against a new alias when the –checkpattern is encountered; if there were two patterns, each pattern would be tried to the checkpattern sample when seen:

```
 java -jar vw4tools.jar --pattern="([^-]+)-[^-]+$" --patt
"NetApp-123456" --> "NetApp"
"NetApp-123456" --> "NetApp-123456"
"UberServer_44_HBA0" --> "UberServer_44_hba0"
"UberServer_44_HBA0" --> "UberServer_44"
```

in this example, two patterns ("([^-]+)-[^-]+$" and "^(.*)_hba(\d+)$") are loaded; when "NetApp-123456" is given as a checkpattern, each is tried in turn, so we see that "([^-]+)-[^-]+$" converts "NetApp-123456" to "NetApp", which means a bunch of devices "NetApp-123456", "NetApp-123457", "NetApp-ABCDEF" would be child entities of a larger entity "NetApp". In essence, all the NetApp-* are collected into the same contailer called "NetApp". Similarly, "([^-]+)-[^-]+$" and "^(.*)_hba(\d+)$"

are tested against "UberServer_44_hba0" when the checkpattern is given, and we can see that "([^-]+)-[^-]+$" makes no difference to the value, but "^(.*)_hba(\d+)$" has the desired effect of chopping off the "_hba#" portion.

Chapter 8. Class Documentation

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

Classes

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

- static Integer compatibilityVersion

  singleton-used compatibility: if (re)set to null, _compatibilityVersion() recalculates

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

- static int _compatibilityVersion ( )

- static String compatibilityVersion ( )
- static String compatibilityVersion (String newVersion)

- 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 ( )

- void maybeAdopt (Entity e)
- String name ( )
- abstract Entity newParent (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

static int _compatibilityVersion ()

calculate the compatibilityVersion as required

Returns: . integer value of compatibilityVersion

JVM Options

compat.target (values: X.Y.Z as a version. For example: 4.0.1, 4.1, 4, 4.2) can be used to tell the JSON output to use specific features available on later versions of the VirtualWisdom4 product. Initially controls whether fcport entities are created but may expand.

static String compatibilityVersion ()

convert the compatibilityVersion into a string

Returns: . string value of compatibilityVersion

static String compatibilityVersion (String newVersion)

set a new compatibility string to set a different target version

This, with compatibilityVersion() and _compatibilityVersion(), is a lot of scaffolding around compatibility outputs; I'm both trying to give some versatility here, plus experimenting myself with properties-vs-commandlines-vs-both. At the end of the day, both methods work to set a version for output, initially a

determiner whether fcport entities are produced. There's future possibility for growth, including config/properties files, afforded by this massive 4-part scaffolding with delusions of grandeur.

Table 8.1. Parameters

newVersion	the new value to use as a version

Returns: . previous config version

abstract boolean canBeChild (Entity e)

whether a given entity can be this entity's child

Table 8.2. Parameters

e	entity to check for possible descendent-hood

Returns: . true if this entity accepts children of "e"'s descendent type

Vector<Entity> children ()

get a list of child entities (local access to local singleton .children)

Returns: . list of zero or more child entities but never null

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 8.3. Parameters

tag	default tag to apply

Entity (String name)

Class Constructor with no initial child.

Table 8.4. Parameters

name	initial name of the new entity

Entity (String name, Entity e)

Class Constructor with an initial child to absorb.

Table 8.5. Parameters

name	initial name of the new entity
e	Entity to consider for adoption as child

String description ()

the description of the entity showing source

Returns: . description of the entity

void maybeAdopt (Entity e)

Convenience function: Entity should either adopt a given child "e" or throw an exception.

This allows very simple coding of streamlining the adoption in a cleaner iteration loop.

Table 8.6. Parameters

e	Entity to consider for adoption as child

String name ()

unique name of the entity: getter for internal variable

Returns: . the name

abstract Entity newParent (String name)

create a new Entity of the correct class to be a parent of this one.

This function is used to polymorphically create the parentage of an entity such as the host holding an HBA

Returns: . new parent for this entity

Table 8.7. Parameters

name	initial name of the new entity

void setDescription (String description)

set the description of the entity to show its source: setter pattern

Table 8.8. Parameters

description	the description to set

void setname (String name)

set the unique name of the entity: setter

Table 8.9. Parameters

name	the name to set

int addTo (VWImport v, String tag)

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

Table 8.10. 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::EntityArray class Reference

- EntityArray (String name, Entity e)
- Entity newParent (String name)
- boolean canBeChild (Entity e)
- org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

Brief Description

An EntityArray is the representation of an Array 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 EntityArray.java

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

- EntityArray.java

EntityArray (String name, Entity e)

Basic Class Constructor.

Entity newParent (String name)

create a new Entity of the correct class to be a parent of this one

boolean canBeChild (Entity e)

whether a given entity can be this entity's child

Returns: . true if accepted, false if refused

Table 8.11. Parameters

e	entity to check for possible descendent-hood

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 8.12. Parameters

tag	default tag to apply

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

- EntityFA (String name, String wwn)
- Entity newParent (String name)
- 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.

Entity newParent (String name)

create a new Entity of the correct class to be a parent of this one

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 8.13. Parameters

tag	default tag to apply

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

- EntityHBA (String name, String wwn)
- Entity newParent (String name)
- 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.

Entity newParent (String name)

create a new Entity of the correct class to be a parent of this one

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 8.14. Parameters

tag	default tag to apply

org::smallfoot::vw4::EntityHost class Reference

- EntityHost (String name, Entity e)
- Entity newParent (String name)
- boolean canBeChild (Entity e)
- org.smallfoot.vw4.VWImport.Entity vwentity (String tag)

Brief Description

An EntityHost is the representation of an Host 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 EntityHost.java

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

- EntityHost.java

EntityHost (String name, Entity e)

Basic Class Constructor.

Entity newParent (String name)

create a new Entity of the correct class to be a parent of this one

boolean canBeChild (Entity e)

whether a given entity can be this entity's child

Returns: . true if accepted, false if refused

Table 8.15. Parameters

e	entity to check for possible descendent-hood

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 8.16. Parameters

tag	default tag to apply

org::smallfoot::vw4::VirtualWisdom4ClientTool::EntityS interface Reference

- boolean select (Entity e)

Brief Description

simple convenience interface to allow entity selection to be codified and created on-the-fly in a query API

boolean select (Entity e)

function to define whether an entity should be selected based on its attributes

Table 8.17. Parameters

e	entity to consider

Returns: . true if this entity satisfies the coded logic

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 124 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 8.18. Parameters

s	description to encapsulate

ImproperChildException (Entity e, Entity p)

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

Table 8.19. 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)
- Entity newParent (String name)
- String parentName ( )
- 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 an initial child to absorb.

Table 8.20. Parameters

name	initial name of the new entity
wwn	initial child entity for this Leaf Entity

Entity newParent (String name)

create a new Entity of the correct class to be a parent of this one: this function is stubbed to return null so that this class can be a static class and directly extensible.

Not really 100% good practice, but the descendency of this LeafEntity is intended to collect functionality. This class is wrapped inside an Entity to retain a clear affinity, but thence needs to be static to be extensible. "She swallowed the cat to catch the spider, she swallowed the spider to catch the fly.. "

Descendents will override this method

Returns: . new parent for this entity

Table 8.21. Parameters

name	initial name of the new entity

String parentName ()

convenience function: if this entity has a parent, show the parent's name, otherwise show this entity's name.

Used during OrderedTuples written via VirtualWisdom4ClientTool.writeOrderedTuples(String, EntitySelector), this allows a simpler, consistent coding when OrderedTuples is being exported.

Returns: . name of parent entity is existent, otherwise name of this entity

String wwn ()

the unique WWPN of the hba: getter for internal variable

Returns: . the WWPN

boolean canBeChild (Entity e)

whether a given entity can be this entity's child

Returns: . true if accepted, false if refused. LeafEntity has no children so this method will always be false

Table 8.22. Parameters

e	entity to check for possible descendent-hood

Returns: . true if this entity accepts children of "e"'s descendent type (always false for Leaf Entity)

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

create a bogus function to avoid build errors

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

Classes

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

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)
- void setTarget (String ver)
- 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 report ( )
- void usage (String progname)
- void writeOrderedTuples (String name, EntitySelector sel)
- void writeOutputfile (String filename)

- static void main (String args)
- static String report (int orphans, int fostered, int total)

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 8.23. 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 8.24. Parameters

filename	file to load

Vector<Pattern> patterns ()

String prettyJSON (VWImport v)

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

Table 8.25. 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 8.26. Parameters

filename	filename to save into

void setTarget (String ver)

WWNDescription wwndesc ()

void _load (String filename)

Open a file.

This is actually a wrapper for the underlying file load

Table 8.27. Parameters

filename	file to load

Table 8.28. 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 8.29. 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 8.30. 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 8.31. 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 and execute it immediately on all current orphan LeafEntities.

The pattern given will be absorbed for the side-benefit of a user testing patterns against names using the –checkpattern option. The pattern is also immediately executed, such that the following four cases occur for each orphan entity evaluated:

- the resulting name is not an entity that exists in the permanent nor hold space; create a hold-space entity which adopts this orphan
- the resulting name is an entity which does adopt this orphan; orphan is adopted by named entity
- the resulting name is an entity which does not adopt this orphan; entity does not adopt this one nor does a new entity get created in hold space
- the resulting name matches the current orphan entity; create a hold-space entity which adopts this orphan and suffixes its name

When complete, the "hold" space is evaluated to remove any "parents" for which there is only one child, thus eliminating lame pattern results before any next run

Table 8.32. Parameters

pattern	a String pattern to execute and 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 report ()

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 8.33. Parameters

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

void writeOrderedTuples (String name, EntitySelector sel)

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

Table 8.34. Parameters

name	filename to write
sel	an optional selector-method

Table 8.35. Exceptions

java.io.IOException	when PrintStream throws an exception

void writeOutputfile (String filename)

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

Table 8.36. 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 8.37. Parameters

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

static String report (int orphans, int fostered, int total)

Chapter 9. File Documentation

README.dox File Reference

Detailed Description

Building .

This project is a basic Autotools:

./autoreconf -vfi.

./configure.

make.

This will create you:

a java/vw4tools.jar that is the compiled code form this project.

a convjars/vw4tools.jar file that is all the compiled code plus dependencies.

This convjars/vw4tools.jar is used on self-testing or regression-testing; it's available at http://chickenandpork.github.io/vw4tools/vw4tools.jar if you really need it to solve a problem TODAY

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

EntityArray.java File Reference

Classes

- struct org::smallfoot::vw4::EntityArray

Namespaces

- struct org::smallfoot::vw4

Detailed Description

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

EntityHost.java File Reference

Classes

- struct org::smallfoot::vw4::EntityHost

Namespaces

- struct org::smallfoot::vw4

Detailed Description

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

VirtualWisdom4ClientTool.java File Reference

Classes

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

csv-to-json.awk File Reference

Detailed Description

Definition in file /Users/allan.clark/src/vw4tools/scripts/csv-to-json.awk

Chapter 10. Directory Documentation

java Directory Reference

Files

- file Entity.java
- file EntityArray.java
- file EntityFA.java
- file EntityHBA.java
- file EntityHost.java
- file VirtualWisdom4ClientTool.java
- file VWImport.java

Detailed Description

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

scripts Directory Reference

Files

- file csv-to-json.awk

Detailed Description

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