

**vw4tools**

---

**vw4tools**

---

---

# Table of Contents

1. Namespace Documentation .....	1
com::fasterxml::jackson::core .....	1
com::fasterxml::jackson::core::json .....	1
com::fasterxml::jackson::databind .....	1
com::fasterxml::jackson::databind::jsonschema .....	1
org .....	1
org::smallfoot .....	1
org::smallfoot::vw4 .....	1
2. JavaDoc API Markup for vw4tools .....	2
3. README .....	3
4. Todo List .....	4
5. Class Documentation .....	5
org::smallfoot::vw4::Entity class Reference .....	5
abstract boolean canBeChild (Entity e) .....	6
Vector<Entity> children () .....	6
abstract org.smallfoot.vw4.VWImport.Entity ventity (String tag) .....	6
Entity (String name) .....	6
Entity (String name, Entity e) .....	6
String description () .....	7
String name () .....	7
void setDescription (String description) .....	7
void setname (String name) .....	7
int addTo (VWImport v, String tag) .....	7
boolean isOrphan () .....	7
boolean isOrphan (TreeMap< String, Entity > list) .....	7
org::smallfoot::vw4::EntityFA class Reference .....	7
EntityFA (String name, String wwn) .....	8
String wwn () .....	8
boolean canBeChild (Entity e) .....	8
org.smallfoot.vw4.VWImport.Entity ventity (String tag) .....	8
org::smallfoot::vw4::EntityHBA class Reference .....	8
EntityHBA (String name, String wwn) .....	9
String wwn () .....	9
boolean canBeChild (Entity e) .....	9
org.smallfoot.vw4.VWImport.Entity ventity (String tag) .....	9
java::lang::Exception class Reference .....	9
org::smallfoot::vw4::Entity::ImproperChildException class Reference .....	9
ImproperChildException (String s) .....	10
ImproperChildException (Entity e, Entity p) .....	10
org::smallfoot::vw4::VirtualWisdom4ClientTool class Reference .....	10
VirtualWisdom4ClientTool (String xmlFile) .....	12
VirtualWisdom4ClientTool () .....	12
TreeMap<String,Entity> entities () .....	12
void load (String filename) .....	12
Vector<Pattern> patterns () .....	12
String prettyJSON (VWImport v) .....	12
void save (String filename) .....	12
WWNDescription wwndesc () .....	13
void _load (String filename) .....	13
void _save (String filename) .....	13
void loadAndAbsorbFile (String f) .....	13

VWImport vwimport () .....	13
void absorbPattern (String pattern) .....	13
Entity entityAccepting (Entity newguy) .....	14
String [] exercisePattern (String alias) .....	14
int finalizeEntities () .....	14
void usage (String progname) .....	14
void writeOrderedTuples (String name) .....	14
void writeOutputfile (String filename) .....	14
static void main (String args[]) .....	15
6. File Documentation .....	16
README.dox File Reference .....	16
Entity.java File Reference .....	16
EntityFA.java File Reference .....	16
EntityHBA.java File Reference .....	17
VirtualWisdom4ClientTool.java File Reference .....	17
VWImport.java File Reference .....	17
7. Directory Documentation .....	18
java Directory Reference .....	18

---

## List of Tables

5.1. Parameters .....	6
5.2. Parameters .....	6
5.3. Parameters .....	7
5.4. Parameters .....	8
5.5. Parameters .....	9
5.6. Parameters .....	10
5.7. Parameters .....	10
5.8. Parameters .....	12
5.9. Parameters .....	12
5.10. Parameters .....	12
5.11. Parameters .....	12
5.12. Parameters .....	13
5.13. Exceptions .....	13
5.14. Parameters .....	13
5.15. Parameters .....	13
5.16. Parameters .....	14
5.17. Parameters .....	14
5.18. Exceptions .....	14
5.19. Parameters .....	15
5.20. Parameters .....	15

---

# Chapter 1. Namespace Documentation

**com::fasterxmlxml::jackson::core**

**com::fasterxmlxml::jackson::core::json**

**com::fasterxmlxml::jackson::databind**

**com::fasterxmlxml::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

---

## Chapter 3. README



---

## Chapter 4. Todo List

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

---

# Chapter 5. Class Documentation

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

### Classes

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

### 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 vwentify ( 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 iomodels 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**

**Protected Attributes**

- String wwn  
*the unique WWPN of the hba*
- EntityFA ( String name, String wwn)
- String wwn ()
- boolean canBeChild ( Entity e)
- org.smallfoot.vw4.VWImport.Entity ventity ( 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)

*Class Constructor with no initial child.*

### String wwn ()

### boolean canBeChild (Entity e)

## org.smallfoot.vw4.VWImport.Entity ventity (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

### Protected Attributes

- String wwn

*the unique WWPN of the hba*

- EntityHBA ( String name, String wwn)

- String wwn ()

- boolean canBeChild ( Entity e)

- org.smallfoot.vw4.VWImport.Entity ventity ( 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)

*Class Constructor with no initial child.*

### String wwn ()

### boolean canBeChild (Entity e)

### 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::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)
- WWNDDescription wwndesc ( )
- void \_load ( String filename)
- void \_save ( String filename)
- void loadAndAbsorbFile ( 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 56 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/>

## VWlimport 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 progame)

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

**Table 5.16. Parameters**

progame	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 5.17. Parameters**

name	filename to write
------	-------------------

**Table 5.18. Exceptions**

java.io.IOException	when PrintStream throws an exception
---------------------	--------------------------------------

### void writeOutputfile (String filename)

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

**Table 5.19. 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 5.20. 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`

### 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/