Mobile Objects and Agents (MOA)
Design Specification

Dejan Milojicic, Bill Laforge and Deepika Chauhan,

The Open Group, Research Institute
11 Cambridge Center
Cambridge, MA 02142
[d.milojicic, d.chauhan, b.laforge]@osf.org

Version 1.2

1 Introduction

The Design Specification is the third specification related to the Mobile Objects and Agents (MOA) project. It is preceded by the Architectural Specification and Functional Specification.

1.1 Purpose of the Document

The purpose of this document is to provide detailed design for the MOA project. This document should be read in conjunction to the APIs produced by JavaDoc from the source code. The APIs contain detailed description of all the interfaces, classes and exceptions. In order to avoid any duplication with the javadoc-produced APIs, this document does not repeat description of each particular parameter, field, etc. Rather, it provides additional description of the interfaces and classes and attempts to provide a broader perspective on the whole agent system. It classifies system into subsystems, and describe the packages/interfaces/classes that build each subsystem.

1.2 Scope of the System

The scope of the system is Internet and intranets.

1.3 Definitions, Acronyms and Abbreviations

For additional details on definitions, acronyms and abbreviations, see Section 1.3 in Functional Specification.

1.4 Naming Standards

In accordance with standard Java naming conventions, we shall name our packages in the following form:

ORG.opengroup.moa.<packageName>

This way we shall distinguish from the packages produced by others in TOG. Beside RI, there is/will be other packages produced at either OSF, X/OPEN or other organizations that might join TOG in the future, hence we define a third level. The "moa" defines our MOA package.

1.5 Overview of the document

Section 2 contains references. Section 3 provides decomposition description. Section 4 to Section 21 describevarious MOA packages.

2 References

This document expands on the three preceding documents:

Milojicic, D. "Mobile Agents, Implementation Plan", The Open Group Research Institute, available off of the MOA Web page: http://www.osf.org/RI/DMO/dmo.htm (All subsequent OSF RI documents are also available from the same location).

Milojicic, D., et. al., "Mobile Objects and Agents, Architectural Specification", The Open Group Research Institute.

Milojicic, D., et. al., "Mobile Objects and Agents, Functional Specification", The Open Group Research Institute.

For additional references, see Section 1.4 in Functional Specification. and Bibliographies in both Functional and Architectural Specification.

3 Decomposition Description

3.1 Overview

The design of the MOA system is presented in Figure 1. This section contains partial description consisting of currently implemented modules. As design and implementation progress, the document will expand.

3.2 Entities

The MOA project will consist of the following entities:

Subsystems represent a functional unit composed of multiple packages and/or classes. The MOA agent subsystems are presented in Figure 1. The following subsystems are presented:

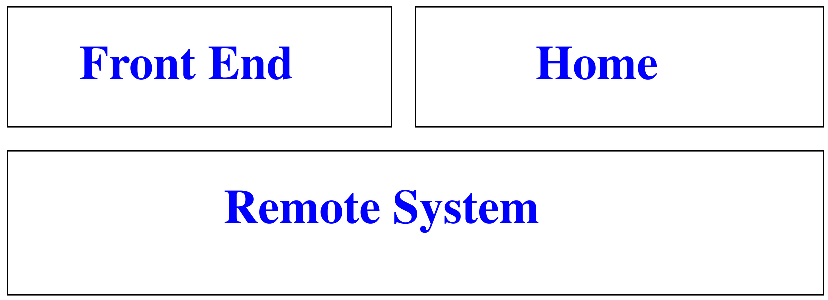

FIGURE 1. **The MOA Subsystems.** The MOA system consists of the following subsystems: (1) Remote System; (2) Front End; and, (3) Home

- **Remote System** subsystem supports agent services. It consists of a Killer, Logger, NameServers, Monitor, AcTakedownMgr, AE, HostPolicy, Messenger and Net. In addition, the AC and the associated agent/place combination also belong to this subsystem
- **Front End** subsystem consists of a BatchDriver running in a separate JVM.
- **Home** subsystem provides home services support.

The following sections will discuss the subsystems in greater detail.

3.3 Front End

This subsystem is responsible for interaction with the MOA system. It consists of MoaApplet, BatchDriver, and Graphical User Interface (GUI). *MoaApplet* provides the connection between the operator and the user object. At start-up, the applet sends a request to the monitor to create the user object. It also supports the launching of pre-configured agents as well as execution of commands. The *BatchDriver*, a batch script, is the command line interface that enables the operator to interact with the moa system. The BatchDriver supports operator logon, agent launching, display of launched and pre-configured agents, etc. . In addition, MOA provides an operator with a *GUI* to the moa system. The operator is able to logon, launch and agent, display lists of pre-configured and launched agents, display and update the agent family, debug using MOA's trace facility. In addition, the operator may also execute commands targeted at an agent.

3.4 Home

Home services provide a respository for snapshots and logs of all agents for which that site serves as a home or an alt-home (alternate home). In addition it also serves as a placeholder for properties of agents at that site. The home service must track which users may access data for which agents as well as which agents may use that site for home service. The home service also supports snapshot retrieval queries The homservice also provides interactive access to agents and places via the Monitor. Agents can be created, killed, suspended, resumed, searched for, etc. Places can be inspected for residing agents.

3.5 Remote System

Remote system represents a fully configured system that agents can visit. In the working environment, it is expected that many such systems will exist. Remote system can also play the role of the home services system, or host some of the front end functionality (although the latter is not an expected configuration). Therefore, remote system contains most of the components of the MOA system. The main components are agent environment together with agent control, agent and place representing each agent, communication stack consisting of the Net package, Messenger and LocatorUser, name server, policies for the remote system, as well as per agent, etc.

4 Awt

The ORG.opengroup.awt package contains classes that provide some useful GUI widgets that are used in the various frontends in the MOA system. eg. Moajar.

4.1 ButtonDialog

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Window
                           |
                           +----java.awt.Dialog
                                   |
                                   +----ORG.opengroup.awt.ButtonDialog
```

public class ButtonDialog extends Dialog

A Dialog which displays a row of buttons.

4.1.1 Constructor Index

ButtonDialog(Frame, String, int, int, String) Construct the dialog.

4.1.2 Method Index

getSelection() Returns the button selected, starting at 0.

processWindowEvent(WindowEvent)

4.1.3 Constructors

ButtonDialog public ButtonDialog(Frame parent, String title, int x, int y, String buttonLabels)

Construct the dialog. To allow subclassing, show() is not called.

Parameters:

parent - The parent frame.

title - Window title.

x - x-location of the dialog.

y - y-location of the dialog.

buttonLabels - Button labels, one button per line.

4.1.4 Methods

protected void processWindowEvent(WindowEvent e)

Overrides:

processWindowEvent in class Window

public int getSelection()

Returns the button selected, starting at 0. If the dialog was closed without a selection, -1 is returned.

4.2 MsgFieldDialog

```
java.lang.Object
  |
  +----java.awt.Component
         |
         +----java.awt.Container
                |
                +----java.awt.Window
                       |
                       +----java.awt.Dialog
                              |
                              +----ORG.opengroup.awt.ButtonDialog
                                     |
                                     +----ORG.opengroup.awt.TextAreaDialog
                                            |
                                            +----ORG.opengroup.awt.MsgFieldDialog
```

public class MsgFieldDialog extends TextAreaDialog

A Dialog which displays a row of buttons below a text area and a text field.

4.2.1 Constructor Index

MsgFieldDialog(Frame, String, int, int, String, String, int, int, String)

Construct the dialog.

4.2.2 Method Index

getText() Returns the current value of the text field.

4.2.3 Constructors

MsgFieldDialog public MsgFieldDialog(Frame parent, String title, int x, int y, String txt, String init,

int col, int sel, String buttonLabels)

Construct the dialog. To allow subclassing, show() is not called.

Parameters:

parent - The parent frame.

title - Window title.

x - x-location of the dialog.

y - y-location of the dialog.

txt - Text used to initialized the text area.

init - Text used to initialized the text field.

col - Width of text field.

sel - Selection to be returned by the text field.

buttonLabels - Button labels, one button per line.

4.2.4 Methods

public String getText()

Returns the current value of the text field.

4.3 TextAreaDialog

```
java.lang.Object
  |
  +----java.awt.Component
          |
          +----java.awt.Container
                  |
                  +----java.awt.Window
                          |
                          +----java.awt.Dialog
                                  |
                                  +----ORG.opengroup.awt.ButtonDialog
                                          |
                                          +----ORG.opengroup.awt.TextAreaDialog
```

public class TextAreaDialog extends ButtonDialog

A Dialog which displays a row of buttons below a non-editable text area.

4.3.1 Constructor Index

TextAreaDialog(Frame, String, int, int, String, String) Construct the dialog.

4.3.2 Constructors

TextAreaDialog public TextAreaDialog(Frame parent, String title, int x, int y, String txt, String buttonLabels)

Construct the dialog. To allow subclassing, show() is not called.

Parameters:

parent - The parent frame.

title - Window title.

x - x-location of the dialog.

y - y-location of the dialog.

txt - Text used to initialized the text area.

buttonLabels - Button labels, one button per line.

4.4 TopicsDialog

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Window
                           |
                           +----java.awt.Dialog
                                   |
                                   +----ORG.opengroup.awt.ButtonDialog
                                           |
                                           +----ORG.opengroup.awt.TopicsDialog
```

public class TopicsDialog extends ButtonDialog

Help Topics Dialog. Driven by a file of topics and file names, the user is presented with a list of topics, each of which has an associated text file.

4.4.1 Constructor Index

TopicsDialog(Frame, int, int, Object)

4.4.2 Constructors

TopicsDialog public TopicsDialog(Frame parent, int x, int y, Object pobj)

Parameters:

parent - The parent frame.

x - x-location of the dialog.

y - y-location of the dialog.

pobj - The object whose class determines the location of the related help files.

5 Beans

The ORG.opengroup.beans contains classes that assist in the manipulation and transfer of beans that are in jar files. It provides classes that are used as containers for jar files and provide the ability to transfer jar files as ascii files.

5.1 Debug

```
java.lang.Object
   |
   +----ORG.opengroup.beans.Debug
```

public final class Debug extends Object implements Serializable

This bean supports debugging by class.

5.1.1 Constructor Index

Debug() Default initialization

5.1.2 Method Index

getClass(int) The name of the class type of the object at index i

setSize() The number of objects being debugged

isDebug(Class) Check if a class wants to be debugged

isDebug(Object) Check if a class wants to be debugged

isDebug(String) Check if a class wants to be debugged

seClass(int, String) The class type at index i in the vector containing the Debug classes

setSize(int) The total number of objects that can be debugged

5.1.3 Constructors

Debug public Debug()

Default initialization

5.1.4 Methods

public static boolean isDebug()

Check if a class wants to be debugged

Parameters:

c - the class type

Returns:

true is the class is registered for debugging false otherwise

public static boolean isDebug(String name)

Check if a class wants to be debugged

Parameters:

s - the component's name t

Returns:

true is the class is registered for debugging false otherwise

public static boolean isDebug(Object self)

Check if a class wants to be debugged

Parameters:

self - the object to be debugged

Returns:

true is the class is registered for debugging false otherwise

public int getSize()

The number of objects being debugged

public void setSize(int s)

The total number of objects that can be debugged

public String getClass(int i) throws ArrayIndexOutOfBoundsException

The name of the class type of the object at index i

public void setClass(int i, String s) throws ArrayIndexOutOfBoundsException

The class type at index i in the vector containing the Debug classes

5.2 EditorPath

```
java.lang.Object
   |
   +----ORG.opengroup.beans.EditorPath
```

public class EditorPath extends Object

Provides access to the list of packages containing property editors.

5.2.1 Constructor Index

EditorPath()

5.2.2 Method Index

getPath() Return editor path as blank-delimited list of package names.

setPath(String) Set editor path to blank-delimited list of package names.

5.2.3 Constructors

EditorPath public EditorPath()

5.2.4 Methods

public static void setPath(String v)

Set editor path to blank-delimited list of package names.

public static String getPath()

Return editor path as blank-delimited list of package names.

5.3 Main

```
java.lang.Object
   |
   +----ORG.opengroup.beans.Main
```

public class Main extends Object

The Main class is a very simple bean container which instantiates a bean using java.beans.Beans.instantiate. Main takes one command line parameter, the name of the bean to be instantiated.

5.3.1 Constructor Index

Main()

5.3.2 Method Index

main(String[]) Program starting point.

5.3.3 Constructors

Main public Main()

5.3.4 Methods

public static void main(String args[])

Program starting point.

5.4 Naked

```
java.lang.Object
   |
   +----ORG.opengroup.beans.Naked
```

public class Naked extends Object

Naked is a static class which supports Beans stored in the directory tree starting at the current working directory.

Naked requires that the CLASSPATH begins with ".". Otherwise new beans might not override old beans found elsewhere on the CLASSPATH.

5.4.1 Constructor Index

Naked()

5.4.2 Method Index

delete(String) Delete bean.

exists(String) Check if bean exists in the naked directory tree.

isNaked() Return true if the CLASSPATH begins with ".".

load(String) Load a Bean.

mkdirs(String) Create directory from package name.

rename(String, String) Rename a bean.

save(Object, String) Save a Bean as a disk file in the directory tree starting at the current working directory.

5.4.3 Constructors

Naked public Naked()

5.4.4 Methods

public static boolean isNaked()

Return true if the CLASSPATH begins with ".". Unless this requirement is meet, new beans may not correctly override older beans found elsewhere on the CLASSPATH.

public static Object load(String name) throws IOException, ClassNotFoundException

Load a Bean. (Uses java.beans.Beans.instantiate.) The name of the bean must be of the form x.y.Z, where x.y is the full package name and Z is the class name.

Parameters:

name - Name of the class or bean

Returns:

the instantiated object

Throws: ClassNotFoundException

if the class of the bean could not be found

Throws: IOException

if an I/O error occurs

public static boolean mkdirs(String pck)

Create directory from package name.

Parameters:

pck - Package name

Returns:

True if the directory exists or was created

public static boolean save(Object obj, String name) throws IOException

Save a Bean as a disk file in the directory tree starting at the current working directory. The name of the bean must be of the form x.y.Z, where x.y is the full package name and Z is the class name. The file saved under the name x/y/Z.ser (or x\y\Z.ser).

Parameters:

obj - Object to be saved

name - Name of the object

Returns:

True if the object was saved

Throws: IOException

if an I/O error occurs

public static boolean rename(String oldNam, String newNam)

Rename a bean. The names must be in the form x.y.Z, where x.y is the full package name and Z is the class name.

Parameters:

oldNam - Name of the bean to be renamed

newNam - The new name of the bean

Returns:

True if the rename was successful

public static boolean delete(String name)

Delete bean. The name must be in the form x.y.Z, where x.y is the full package name and Z is the class name.

Parameters:

name - Name of the bean to be deleted.

Returns:

True if the delete was successful.

public static boolean exists(String name)

Check if bean exists in the naked directory tree. The name must be in the form x.y.Z, where x.y is the full package name and Z is the class name.

Parameters:

name - Name of the bean to be deleted.

Returns:

True if the delete was successful.

5.5 PropertyEditorAccess

```
java.lang.Object
  |
  +----ORG.opengroup.beans.PropertyEditorAccess
```

public class PropertyEditorAccess extends Object

Determines if a property is Read/Write and has an associated property editor. Supports indexed properties.

5.5.1 Constructor Index

PropertyEditorAccess(Object) Select the object to be edited.

5.5.2 Method Index

getBeanInfo() Returns the BeanInfo of the object to be edited.

getIndexedPropertyEditor() Returns the Indexed Property Editor of the selected property, or null.

getPropertyDescriptor() Returns the Property Descriptor of the selected property, or null.

getPropertyDescriptors() Returns the Property Descriptors of the object to be edited.

getPropertyEditor() Returns the Property Editor of the selected property, or null.

getValue() Get the current value of the property as a text string.

getValue(int) Get the current value of the indexed property as a text string.

selectProperty(int) Select a property on the object.

selectProperty(String) Select a property on the object.

setValue(int, String) Set the current value of the indexed property as a text string.

setValue(String) Set the current value of the property as a text string.

5.5.3 Constructors

PropertyEditorAccess public PropertyEditorAccess(Object v) throws IntrospectionException, ClassNotFoundException

Select the object to be edited.

Throws: IntrospectionException

    unable to access BeanInfo.

5.5.4 Methods

public BeanInfo getBeanInfo()

Returns the BeanInfo of the object to be edited.

public PropertyDescriptor[] getPropertyDescriptors()

Returns the Property Descriptors of the object to be edited.

public boolean selectProperty(String v)

Select a property on the object. Returns true if the property has an editor.

public boolean selectProperty(int ndx)

Select a property on the object. Returns true if the property has an editor.

public PropertyDescriptor getPropertyDescriptor()

Returns the Property Descriptor of the selected property, or null.

public PropertyEditor getPropertyEditor()

Returns the Property Editor of the selected property, or null.

public PropertyEditor getIndexedPropertyEditor()

Returns the Indexed Property Editor of the selected property, or null.

public String getValue() throws IllegalAccessException, InvocationTargetException

Get the current value of the property as a text string. If null is returned, then the property can not be edited as a string.

public void setValue(String v) throws IllegalAccessException, InvocationTargetException

Set the current value of the property as a text string.

public String getValue(int n) throws IllegalAccessException, InvocationTargetException, ArrayIndexOutOfBoundsException

Get the current value of the indexed property as a text string. If null is returned, then the property can not be edited as a string.

public void setValue(int n, String v) throws IllegalAccessException, InvocationTargetException, ArrayIndexOutOfBoundsException

Set the current value of the indexed property as a text string.

5.6 Ready

```
java.lang.Object
  |
  +----ORG.opengroup.beans.Ready
```

public class Ready extends Object implements Serializable

Ready will call method deserialized when the object is deserialized.

The intent here is that a Bean will have a nested class which extends Ready, overriding the deserialized method. In the following example example, "Blip!" us printed when TestReady is deserialized:

```
public class TestReady
 {
    Ready ready = new Ready()
    {
       public void deserialized()
       {
          System.out.println("Blip!");
       }
    };
```

}

5.6.1 Constructor Index

Ready()

5.6.2 Method Index

deserialized()

5.6.3 Constructors

Ready

public Ready()

5.6.4 Methods

public void deserialized()

5.7 ReadyThread

```
java.lang.Object
  |
  +----java.lang.Thread
          |
          +----ORG.opengroup.beans.ReadyThread
```

public class ReadyThread extends Thread implements Serializable

ReadyThread starts running when an instance is constructed and when an instance is deserialized. However, ReadyThread does not start running at “design time".

The intent here is that a Bean will have a nested class which extends ReadyThread, overriding the run method. For example:

```
public class TestReadyThread implements Serializable
 {
    Thread runner= new ReadyThread()
    {
       public void run()
       {
          System.out.println("Blip!");
       }
    };
 }
```

5.7.1 Constructor Index

ReadyThread()

5.7.2 Constructors

ReadyThread public ReadyThread()

5.8 StringArray

```
java.lang.Object
   |
   +----ORG.opengroup.beans.StringArray
```

public class StringArray extends Object implements Serializable

This is a simple bean with property string which is an array of String.

5.8.1 Field Index

v

5.8.2 Constructor Index

StringArray()

5.8.3 Method Index

getSize() The size of the array

getString() The string representation of the array

getString(int) Retrieve a strnig from a specific location

setSize(int) The size of the array

setString(int, String) The string at a specific location in the array

setString(String[]) Initialize the array with these values

5.8.4 Fields

v

protected Vector v

5.8.5 Constructors

StringArray public StringArray()

5.8.6 Methods

public int getSize()

The size of the array

public void setSize(int s)

The size of the array

public String[] getString()

The string representation of the array

public void setString(String a[])

Initialize the array with these values

public String getString(int i) throws ArrayIndexOutOfBoundsException

Retrieve a string from a specific location

public void setString(int i, String s) throws ArrayIndexOutOfBoundsException

The string at a specific location in the array

6 Jar

This package contains classes that assist in the manipulation and transfer of beans that are in jar files. It provides classes that are used as containers for jar files and provide the ability to transfer jar files as ascii files.

6.1 Base64

```
java.lang.Object
   |
   +----ORG.opengroup.jar.Base64
```

public class Base64 extends Object

Base64 is a standard for passing binary data as ASCII.

6.1.1 Constructor Index

Base64()

6.1.2 Method Index

binToAsc(byte[]) Convert an array of binary bytes to an array of ASCII bytes.

6.1.3 Constructors

Base64 public Base64()

6.1.4 Methods

public static byte[] binToAsc(byte bin[])

Convert an array of binary bytes to an array of ASCII bytes.

ASCII characters used are A-Z, a-z, 0-9, /, and +. The '=' is used to pad the ascii to a multiple of 4 bytes.

Parameters:

bin - The binary data.

Returns:

The equivalent ASCII data.

6.2 Bucket

```
java.lang.Object
  |
  +----ORG..Bucket
```

public class Bucket extends Object implements Cloneable, Serializable

Holds the contents of a jar file.

6.2.1 Field Index

manifest The manifest.

MANIFEST_NAME File name of the manifest.

6.2.2 Constructor Index

Bucket() Creates a Bucket with no content.

Bucket(InputStream) Creates a bucket with the content of a jar file.

6.2.3 Method Index

clone() Clones the Bucket.

get(String) Return the byte array associated with the given file name.

getClassLoader() Returns the ClassLoader

getManifest() Returns the manifest.

getObject(String) Either deserialize an existing object or, if there is no such object, create a new object using the null constructor of the class with the same name.

hasConflict(Bucket) Determines if two Buckets have common files which differ in their manifests.

jar(OutputStream) Write the bucket in jar file format (compressed)

modified(String) Return the time when an entry was last modified.

put(String, byte[]) Associates a name and a value.

putObject(String, Object) Serialize an object and save it as a named resource file in the jar.

remove(String) Remove a file and its manifest entry from the bucket.

removeObject(String) Remove an object and its manifest entry from the bucket.

resources() Return an enumeration of the file names.

setNested(Bucket) Assign the bucket in which this bucket is nested.

store(OutputStream) Write the bucket in jar file format (uncompressed)

update(Bucket) Update the bucket with the contents of another.

validate() Validate the jar contents against the manifest security hashes.

6.2.4 Fields

MANIFEST_NAME protected static final String MANIFEST_NAME

File name of the manifest.

manifest protected Manifest manifest

The manifest.

6.2.5 Constructors

Bucket public Bucket()

Creates a Bucket with no content.

Bucket public Bucket(InputStream is) throws IOException, ZipException

Creates a bucket with the content of a jar file.

Parameters:

is - The input stream for the jar file.

Throws: IOException

May be thrown when reading from the input stream.

Throws: ZipException

May be thrown when processing the contents of the jar.

6.2.6 Methods

public void setNested(Bucket context)

Assign the bucket in which this bucket is nested.

public final ClassLoader getClassLoader()

Returns the ClassLoader

public synchronized void jar(OutputStream os) throws IOException, ZipException

Write the bucket in jar file format (compressed)

Parameters:

os - The output stream for the jar file.

Throws: IOException

May be thrown when writting to the output stream.

Throws: ZipException

May be thrown when processing the contents of the jar.

public synchronized void store(OutputStream os) throws IOException, ZipException

Write the bucket in jar file format (uncompressed)

Parameters:

os - The output stream for the jar file.

Throws: IOException

May be thrown when writing to the output stream.

Throws: ZipException

May be thrown when processing the contents of the jar.

public synchronized Object clone()

Clones the Bucket.

Overrides:

clone in class Object

public Object getObject(String name) throws IOException, ClassNotFoundException

Either deserialize an existing object or, if there is no such object, create a new object using the null constructor of the class with the same name.

Parameters:

name - A dot-separated name, with no file extension.

Throws: IOException

Error occurred while deserializing.

Throws: ClassNotFoundException

No object or class by that name.

public synchronized byte[] get(String name)

Return the byte array associated with the given file name.

Parameters:

name - A /-seperated path name, complete with file extensions.

Returns:

The associated byte array or null.

public synchronized long modified(String name)

Return the time when an entry was last modified.

Parameters:

name - A /-separated path name, complete with file extensions.

Returns:

The time in milliseconds, or 0.

public synchronized void putObject(String name, Object obj) throws IOException

Serialize an object and save it as a named resource file in the jar. This is the exact reverse of java.beans.Beans.instantiate. Security hashes are included in the manifest.

Parameters:

name - A dot-separated name, with no file extension. The '.'s are replaced by '/' and ".ser" is appended.

obj - Object to be saved.

Throws: IOException

Error occurred while adding the object.

See Also:

instantiate

public void put(String name,.byte value[])

Associates a name and a value. Security hashes are included in the manifest.

Parameters:

name - A /-separated path name, complete with file extensions.

value - The associated byte array.

public Enumeration resources()

Return an enumeration of the file names.

public void removeObject(String name)

Remove an object and its manifest entry from the bucket.

Parameters:

name - A dot-separated path, with NO file extension.

public synchronized void remove(String name)

Remove a file and its manifest entry from the bucket.

Parameters:

name - A /-separated path name, complete with file extensions.

public synchronized Manifest getManifest()

Returns the manifest.

public synchronized boolean validate() throws NoSuchAlgorithmException

Validate the jar contents against the manifest security hashes. Returns true if all are valid.

public synchronized boolean hasConflict(Bucket b)

Determines if two Buckets have common files which differ in their manifests. This is a very strict test which will detect even differences in the delimiters used in the manifests. It is this strict to meet security requirements. Note hat for every file in a bucket there is a manifest section which contains a security hash for that file, so as long as both buckets have been validated, one need only compare the manifests. Returns true if there is a conflict.

public synchronized void update(Bucket b)

Update the bucket with the contents of another.

6.3 JarFile

```
java.lang.Object
   |
   +----ORG.opengroup.jar.Bucket
           |
           +----ORG.opengroup.jar.JarFile
```

public class JarFile extends Bucket

Adds file operations to Bucket

6.3.1 Constructor Index

JarFile() Creates a Bucket with no content.

JarFile(String) Creates a bucket with the content of a jar file.

6.3.2 Method Index

putFile(String) Put a file in the bucket.

putFile(String, String) Put a file in the bucket.

save() Update the jar file.

saveAs(String) Write the bucket to a jar file.

6.3.3 Constructors

JarFile public JarFile()

Creates a Bucket with no content.

JarFile public JarFile(String pfileName) throws IOException, ZipException

Creates a bucket with the content of a jar file.

Parameters:

pfileName - Name of jar file.

Throws: IOException

May be thrown when reading from the jar file.

Throws: ZipException

May be thrown when processing the contents of the jar file.

6.3.4 Methods

public void saveAs(String pfileName) throws IOException, ZipException

Write the bucket to a jar file.

Parameters:

pfileName - Name of jar file.

Throws: IOException

May be thrown when writting to the jar file.

Throws: ZipException

May be thrown when processing the contents of the jar file.

public void save() throws IOException, ZipException

Update the jar file.

Throws: IOException

May be thrown when writing to the jar file.

Throws: ZipException

May be thrown when processing the contents of the jar file.

public void putFile(String pfileName) throws IOException

Put a file in the bucket.

Parameters:

pfileName - Name of the file to be put in the bucket.

Throws: IOException

May be thrown when reading the file.

public void putFile(String pfileName, String as) throws IOException

Put a file in the bucket.

Parameters:

pfileName - Name of the file to be read.

as - Name of the file added to the bucket.

Throws: IOException

May be thrown when reading the file.

6.4 Manifest

```
java.lang.Object
  |
  +----ORG.opengroup.jar.Manifest
```

public class Manifest extends Object implements Serializable, Cloneable

Represents a manifest of a jar file.

6.4.1 Constructor Index

Manifest() Manifest(byte[])

Create a manifest from an array of bytes.

Manifest(String[]) Create a manifest from an array of strings.

6.4.2 Method Index

add(ManifestSection) Adds the manifest section, unless the name starts with META-INF/ or /META-INF/

clear() Empty the manifest.

clone() Clones the manifest.

force(String) Returns the named manifest section.

get() Returns the array of strings comprising the manifest.

get(String) Returns the named manifest section.

getBytes() Returns the manifest as an array of bytes.

getFull() Returns the array of strings comprising the manifest.

remove(String) Removes the named manifest section.

sections() Returns an enumeration of the sections of the manifest.

6.4.3 Constructors

Manifest public Manifest()

Manifest public Manifest(String txt[]) Create a manifest from an array of strings.

Manifest public Manifest(byte data[]) Create a manifest from an array of bytes.

6.4.4 Methods

public synchronized Object clone()

Clones the manifest.

Overrides:

 clone in class Object

public synchronized void clear()

Empty the manifest.

public synchronized String[] get()

Returns the array of strings comprising the manifest.

public synchronized String[] getFull()

Returns the array of strings comprising the manifest. (Line delimiters are included.)

public byte[] getBytes()

Returns the manifest as an array of bytes.

public synchronized ManifestSection get(String name)

Returns the named manifest section. Null is returned if there is no matching section.

public synchronized ManifestSection force(String name)

Returns the named manifest section. The section is created if necessary.

public synchronized void remove(String name)

Removes the named manifest section.

public synchronized void add(ManifestSection ms)

Adds the manifest section, unless the name starts with META-INF/ or /META-INF/

public Enumeration sections()

Returns an enumeration of the sections of the manifest.

6.5 ManifestSection

```
java.lang.Object
   |
   +----ORG.opengroup.jar.ManifestSection
```

public class ManifestSection extends Object implements Serializable, Cloneable

Represents one section of a manifest from a jar file.

6.5.1 Constructor Index

ManifestSection() Null constructor required for serialization.

ManifestSection(String) Normal constructor initializes the section name.

ManifestSection(String[]) Load a pre-existing manifest section.

6.5.2 Method Index

add(String, String) Add an item to the section.

ddHash(byte[]) Add MD5 and SHA security hashes to the section.

base64Hash(String, byte[]) Returns a security hash as a base64 ASCII string.

clone() Clones the section.

equals(Object) Returns true if the section names are equal.

get() Returns the contents of the section as an array of strings.

get(String) Returns the value of an item in the section, or null.

getFull() Returns the contents of the section as an array of strings.

remove(String) Remove an item from the section.

validate(byte[]) Validate the security hash.

6.5.3 Constructors

ManifestSection public ManifestSection()

Null constructor required for serialization.

ManifestSection public ManifestSection(String id)

Normal constructor initializes the section name.

Parameters:

id - Path name.

ManifestSection public ManifestSection(String header[])

Load a pre-existing manifest section. Assumption: Continuation lines have the same line delimiter.

Parameters:

header - An array of strings in the form "name: value", or " value".

6.5.4 Methods

public synchronized Object clone()

Clones the section.

Overrides:

clone in class Object

public synchronized boolean equals(Object o)

Returns true if the section names are equal.

Parameters:

o - Another ManifestSection.

Overrides:

equals in class Object

public synchronized void remove(String name)

Remove an item from the section.

Parameters:

name - Name of the item.

public synchronized void add(String name, String value)

Add an item to the section.

Parameters:

name - Name of the item.

value - Value assigned to the item.

public synchronized String get(String name)

Returns the value of an item in the section, or null.

Parameters:

name - Name of the item.

Returns:

The assigned value.

public synchronized String[] get()

Returns the contents of the section as an array of strings.

public synchronized String[] getFull()

Returns the contents of the section as an array of strings. (Line delimiters are included.)

public static String base64Hash(String algorithm, byte data[]) throws NoSuchAlgorithmException

Returns a security hash as a base64 ASCII string.

Parameters:

algorithm - Security hash algorithm name.

data - The data to be hashed.

Throws: NoSuchAlgorithmException

Unimplimented security hash.

public void addHash(byte data[])

Add MD5 and SHA security hashes to the section.

Parameters:

data - The data associated with the section.

public boolean validate(byte data[]) throws NoSuchAlgorithmException

Validate the security hash.

Parameters:

data - The data associated with the section.

Returns:

True when valid.

Throws: NoSuchAlgorithmException

Unimplemented security hash.

7 misc

The ORG.opengroup.misc package contains some useful classes that help in the programming process. These include data structures and utility classes

7.1 DoubleHashtable

```
java.lang.Object
  |
  +----ORG.opengroup.misc.DoubleHashtable
```

public class DoubleHashtable extends Object

A structure that maintains two hashtables. One is the inverse of the other

7.1.1 Constructor Index

DoubleHashtable() Creates a new DoublehashTable

7.1.2 Method Index

get(Object) Retrieves a (key, value) pair from the structure

getInverse(Object) Retrieves the inverse of a (key, value) pair from the structure

put(Object, Object) Adds a new (key, value) pair to the structure

remove(Object) Removes a (key,value) pair

removeInv(Object) Removes an inverse (key,value) pair

7.1.3 Constructors

DoubleHashtable public DoubleHashtable()

Creates a new DoublehashTable

7.1.4 Methods

public synchronized void put(Object key, Object value)

Adds a new (key, value) pair to the structure

public synchronized Object get(Object key)

Retrieves a (key, value) pair from the structure

public synchronized Object getInverse(Object key)

Retrieves the inverse of a (key, value) pair from the structure

public synchronized Object remove(Object key)

Removes a (key,value) pair

public synchronized Object removeInv(Object key)

Removes an inverse (key,value) pair

7.2 Line

```
java.lang.Object
   |
   +----ORG.opengroup.misc.Line
```

public class Line extends Object implements Enumeration

A structure tp process a multi-line string

7.2.1 Constructor Index

Line(String) Analyze a multi-line string.

7.2.2 Method Index

getHeight() Returns the number of lines in the string.

getWidth() Returns the number of characters in the longest line.

hasMoreElements() Returns true if there is another line.

nextElement() Returns the next line.

7.2.3 Constructors

Line public Line(String multi)

Analyze a multi-line string. Carriage returns are ignored; only line feeds ('\n') are significant.

Parameters:

multi - The multi-line string or null.

7.2.4 Methods

public int getHeight()

Returns the number of lines in the string. Returns 0 only if null was passed to the constructor. Returns 1 only if there were no line feeds passed to the constructor.

public int getWidth()

Returns the number of characters in the longest line.

public boolean hasMoreElements()

Returns true if there is another line.

public Object nextElement()

Returns the next line. (Cast to String.) If there are no more lines, null is returned.

7.3 MLine

```
java.lang.Object
   |
   +----ORG.opengroup.misc.MLine
```

public class MLine extends Object implements Enumeration

Processes a multi-line string. Intended for use with the Manifest file of a jar, lines may end with CR, LF, or CR LF. Line

delimiters are preserved!

7.3.1 Constructor Index

MLine(String) Analyze a multi-line string.

7.3.2 Method Index

empty(String) An empty string is either zero length or begins with CR or LF.

getHeight() Returns the number of lines in the string.

getWidth() Returns the number of characters in the longest line.

asMoreElements() Returns true if there is another line.

nextElement() Returns the next line.

7.3.3 Constructors

MLine public MLine(String multi)

Analyze a multi-line string.

Parameters:

multi - The multi-line string or null.

7.3.4 Methods

public static boolean empty(String s)

An empty string is either zero length or begins with CR or LF.

public int getHeight()

Returns the number of lines in the string. Returns 0 only if null was passed to the constructor. Returns 1 only if there were no line feeds passed to the constructor.

public int getWidth()

Returns the number of characters in the longest line.

public boolean hasMoreElements()

Returns true if there is another line.

public Object nextElement()

Returns the next line. (Cast to String.) If there are no more lines, null is returned.

7.4 Reads

```
java.lang.Object
  |
  +----ORG.opengroup.misc.Reads
```

public final class Reads extends Object

This is a fast variable-length reader.

It reads into a series of buffers, each buffer being twice as large as the last.

At the end, it allocates a single buffer exactly equal in length to the total bytes read. It then copies everything into this final buffer.

7.4.1 Constructor Index

Reads()

7.4.2 Method Index

read(InputStream)

7.4.3 Constructors

Reads

public Reads()

7.4.4 Methods

public static byte[] read(InputStream is) throws IOException

7.5 TwinAsymmetricHashtable

```
java.lang.Object
   |
   +----ORG.opengroup.misc.TwinAsymmetricHashtable
```

public class TwinAsymmetricHashtable extends Object

This class contains two hashtables: ht - contains key Object mapped to a Vector of values inverseHt - is the inverse of ht, in the sense that ht values are the keys of inverseHt and ht keys become the values. But since ht contains a Vector of values the two hashtables are not symmetric.

7.5.1 Constructor Index

TwinAsymmetricHashtable() Creates a new instance

7.5.2 Method Index

get(Object) returns a new (key, value) pair

getInverse(Object) Returns the inverse (key, value) pair

put(Object, Object) Adds a new (key, value) pair

remove(Object) Removes the key (and corresponding values) from ht Removes each value (key for inverseHt) from the inverseHt

removeInv(Object) Removes the inverse (key,value) pair

7.5.3 Constructors

TwinAsymmetricHashtable public TwinAsymmetricHashtable()

Creates a new instance

7.5.4 Methods

public synchronized void put(Object key, Object value)

Adds a new (key, value) pair

public synchronized Object get(Object key)

returns a new (key, value) pair

public synchronized Object getInverse(Object key)

Returns the inverse (key, value) pair

public synchronized Object remove(Object key)

Removes the key (and corresponding values) from ht Removes each value (key for inverseHt) from the inverseHt

public synchronized Object removeInv(Object key)

Removes the inverse (key,value) pair

8 moa.ac

The "ac" package contains all the files which are necessary for the application Agent/Place to communicate with the rest of the MOA system. It has 22 classes. Agent control (AC) is an internal class that represents an interface between agent environment and agent/place. It manages various objects that represent agent system resources (communication channels, agent capabilities, etc.). These objects can be accessed by an agent/place, but they cannot be changed. This class is instantiated when an agent is created at a node or an agent arrives at a node. Upon instantiation it is added to the Agent Environment (AE) of the node. It is the trusted part of MOA and is in MOA domain. It helps agent and place interface with the rest of the MOA components (Net, Messenger,Logger, Monitor, ... etc.) AC comprises of various components: 1. AcChannelMgr (for managing communication over the channels using ORG.opengroup.sandbox.net package) 2. AcMsgUser (for managing messaging using ORG.opengroup.sandbox.message package) ChannelMove and ChannelMoved are objects exchanged over channels prior to agent movement for a 2-way commit and ChannelClose and ChannelClosed are objects exchanged over channels prior to agent takedown to enable graceful channel takedown. SuspendAgentMvmt, SuspendAgentThread, ResumeAgent, and KillAgent classes represent the type of command that can be sent to the agent. AcTakedownMgr is used for managing graceful takedown of an agent

8.1 AC

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
          |
          +----ORG.opengroup.sandbox.environment.Environment
                  |
                  +----ORG.opengroup.moa.ac.AC
```

public class AC extends Environment

8.1.1 Field Index

AGENT_BEAN_NAME Bean Name for the agent

AGENT_POLICY_BEAN_NAME Bean Name for the agent's place

AGENT_PROPERTIES_BEAN_NAME Bean Name for the agent's AgentProperties

BUCKET_BEAN_NAME Bean Name for the JarFile received

FAMILY_BEAN_NAME Bean Name for the agent's family

LOCAL_POLICY_BEAN_NAME Bean Name for the local policy negotiated between agent and host

LOG_SERVICE_BEAN_NAME Bean Name for the agent's LogService

PLACE_BEAN_NAME Bean Name for the agent's place

8.1.2 Constructor Index

AC() Public Constructor

8.1.3 Method Index

acTakedown() AC Takedown processing.

closeOpenChannels(Api) This method closes all the open channels of the source calling the method

endMove() Used to indicate that Agent move will end

endOperation() Used to end the operation

endSnapshot() Used to indicate that Agent snapshot will end

extract(JarFile, Policy) Called by the Mover to extract contents of the received JarFile

fork(int, int, Agent) This method forks a child agent and launches it on same host.

fork(int, int, Agent, AEname) getService(String)

This method returns the Service specified by the pathname

init() init will be called during EnvironMentChild wiring.

log(Api, String, String) This method is called to log activities specific to this Agent/Place

move(AEname, boolean) Called by the agent to move to a new location in the network This method updates the localPolicy and agentPolicy, writes a ChannelMove on all agent channels (through moveAgentChannels), updates the bucket and calls move on the Mover.

openChannel(DestinationName, Api) This method opens a channel of communication for agent/place via the AcChannelMgr class Parameters:

suspendOperation() Used to suspend an operation when a snapshot is in Progress

processCommand(Object) This method is used to process the command received from the user or the Monitor

sendMessage(Api, DestinationName, Object) Agent/Place use this method to send a message to another destination It uses the AcMsgUser class

snapShot(int) This method suspends the agent, captures its state, and sends it to the agent home or alternate home

startMove() Used to indicate that Agent move will begin

startOperation() Starts an operation on this AC and throws an InvalidStateException if the AC is in any of the three

Invalid States (move, snapshot, and takedown)

startSnapshot() Used to indicate that Agent snapshot will begin

startTakedown() Starts the AC takedown

takedownAC(Failure) This method sends a KillAgent command to the Monitor to take down this AC

8.1.4 Fields

public static final String **BUCKET_BEAN_NAME**

Bean Name for the JarFile received

public static final String **AGENT_BEAN_NAME**

Bean Name for the agent

public static final String **AGENT_PROPERTIES_BEAN_NAME**

Bean Name for the agent's AgentProperties

public static final String **PLACE_BEAN_NAME**

Bean Name for the agent's place

public static final String **AGENT_POLICY_BEAN_NAME**

Bean Name for the agent's place

public static final String **LOCAL_POLICY_BEAN_NAME**

Bean Name for the local policy negotiated between agent and host

public static final String **FAMILY_BEAN_NAM**E

Bean Name for the agent's family

public static final String **LOG_SERVICE_BEAN_NAME**

Bean Name for the agent's LogService

8.1.5 Constructors

AC public AC()

Public Constructor

8.1.6 Methods

public void init()

init will be called during EnvironMentChild wiring. This is called when Mover.move() method does ae.addChild(agentName, AC)

Overrides:

init in class Environment

public Failure extract(JarFile jf, Policy lPolicy)

Called by the Mover to extract contents of the received JarFile

public void startOperation() throws InvalidStateException

Starts an operation on this AC and throws an InvalidStateException if the AC is in any of the three Invalid States (move, snapshot, and takedown)

public void endOperation()

Used to end the operation

public synchronized void pendOperation() throws InvalidStateException, InterruptedException

Used to suspend an operation when a snapshot is in Progress

public void startTakedown()

Starts the AC takedown

public void startMove() throws InvalidStateException

Used to indicate that Agent move will begin

public void startSnapshot() throws InvalidStateException

Used to indicate that Agent snapshot will begin

public void endMove()

Used to indicate that Agent move will end

public void endSnapshot()

Used to indicate that Agent snapshot will end

public Failure move(AEname location, boolean placeRemaining)

Called by the agent to move to a new location in the network This method updates the localPolicy and agentPolicy, writes a ChannelMove on all agent channels (through moveAgentChannels), updates the bucket and calls move on the Mover. In case agent moves without leaving the place it sends a KillAgent command to the Monitor

Parameters:

location - new location to move to

placeRemaining - indicates whether to leave the place at the location

Returns:

null if move succeeds, Failure object indicating reason of Failure if move fails

public Failure snapShot(int snapshotNo)

This method suspends the agent, captures its state, and sends it to the agent home or alternate home

Parameters:

snapshotNo - the sequence number of the agent's snapshot

Returns:

null if snapshot procedure is successful, Failure object containing the reason of the Failure if it fails

public Object fork(int noOfClones, int noOfPlaces, Agent cloneAgent)

This method forks a child agent and launches it on same host.

Parameters:

noOfClones - the number of times the forked agent is allowed to clone

noOfPlaces - the number of places the forked agent is allowed to leave

cloneAgent - the cloned agent

Returns:

a String Object indicating the name of the forked agent if the fork succeeds or a Failure object

indicating the reason of Failure

public Object fork(int noOfClones, int noOfPlaces, Agent cloneAgent, AEname dest)

public Object openChannel(DestinationName destName, Api source)

This method opens a channel of communication for agent/place via the AcChannelMgr class Parameters:

Parameters:

destName - name of the destination to which to open the channel to

source - indicates who makes the openChannel Request (agent/ place)

Returns:

AgentChannelAccess if the openChannelRequest succeeds or Failure if the request fails

public void closeOpenChannels(Api source)

This method closes all the open channels of the source calling the method

Parameters:

source - indicates who makes the closeChannel Request (agent/ place)

public Object sendMessage(Api source, DestinationName name, Object content)

Agent/Place use this method to send a message to another destination It uses the AcMsgUser class

Parameters:

source - indicates who makes the openChannel Request (agent/ place)

destName - name of the destination to which to open the channel to

content - the Object to be sent

public Failure processCommand(Object command)

This method is used to process the command received from the user or the Monitor

Parameters:

command - the command Object. It is of three types: SuspendAgentThread, ResumeAgent, or

KillAgent

Returns:

null if command processing is successful, Failure object indicating the reason of Failure if it fails

public final void takedownAC(Failure reason)

This method sends a KillAgent command to the Monitor to take down this AC

Parameters:

reason - the reason for which this AC is being taken down

public void acTakedown()

AC Takedown processing. This method is called in 2 cases: 1. via AC processCommand method (When

Monitor receives a KillAgent command) 2. via run() method of AcTakedownMgr (when AC lifetime on the node expires)

public void log(Api source, String logType, String logData)

This method is called to log activities specific to this Agent/Place

Parameters:

source - indicates who calls this method (agent/place)

logType - the type of log

logData - the information to be logged

public final Service getService(String fullPathName)

This method returns the Service specified by the pathname

Parameters:

fullPathName - the path name of the service

8.2 AcChannelMgr

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.net.NetUser
                   |
                   +----ORG.opengroup.moa.ac.AcChannelMgr
```

public class AcChannelMgr extends NetUser

A bean which manages all channels for the ac

8.2.1 Constructor Index

AcChannelMgr()

8.2.2 Method Index

getOpenChannelTimeout() return open channel timeout

getSendMessageTimeout() Return the send message timeout

init() Initialize.

processRequest(SandboxRequest) Overrides NetUser.processRequest for incoming channels.

setOpenChannelTimeout(int) Set open channel timeout

setSendMessageTimeout(int) Set send message timeout

8.2.3 Constructors

AcChannelMgr public AcChannelMgr()

8.2.4 Methods

public void init()

Initialize.

Overrides:

init in class NetUser

public final void setSendMessageTimeout(int v)

Set send message timeout

public final int getSendMessageTimeout()

Return the send message timeout

public void setOpenChannelTimeout(int tmo)

Set open channel timeout

public int getOpenChannelTimeout()

return open channel timeout

public void processRequest(SandboxRequest channelRequest)

Overrides NetUser.processRequest for incoming channels. Receives the IncomingChannel objects to associated agent/place

Parameters:

channelRequest - the SandboxRequest containing the IncomingChannel from a remote agent/place

Overrides:

processRequest in class NetUser

8.3 AcMsgUser

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
        |
        +----ORG.opengroup.sandbox.message.MessageUser
              |
              +----ORG.opengroup.moa.nameserver.LocatorUser
                    |
                    +----ORG.opengroup.moa.ac.AcMsgUser
```

public class AcMsgUser extends LocatorUser

8.3.1 Constructor Index

AcMsgUser()

8.3.2 Method Index

getClassLoader(Message) This method overrides the MessageUser method to return the JarClassLoader of the Bucket in case message is moa application specific

getTimeout() returns the timeout of sendRequest in milliseconds

incomingRequest(String, String, int, byte, Object) This method overrides the incomingRequest method of the sandbox.messenger.MessageUser class It receives the incoming messages to the associated AC.

init() Called by the Environment wire() method to integrate into it

isReceiverPresent(byte) This method overrides the LocatorUser method and checks that if the message is foragent then is it present

isSuspended() This method returns whether AcMsgUser is suspended or not

resume() This method overrides the method in MessageUser and resumes it

setTimeout(int) sets the timeout of the sendRequest suspend()

This method overrides the method in MessageUser and suspends it

8.3.3 Constructors

AcMsgUser public AcMsgUser()

8.3.4 Methods

public void setTimeout(int tmo)

sets the timeout of the sendRequest

Parameters:

tmo - the timeout in milliseconds

public int getTimeout()

returns the timeout of sendRequest in milliseconds

public void init()

Called by the Environment wire() method to integrate into it

Overrides:

init in class LocatorUser

public Object incomingRequest(String origName, String origHost, int origPort, byte att, Object content)

This method overrides the incomingRequest method of the sandbox.messenger.MessageUser class It receives the incoming messages to the associated AC. Depending upon the attribute of the received message, it takes different courses of action

Parameters:

origName - name of the originator of message

origHost - host of the originator of message

origPort - port of the originator of message

att - attribute of the message

content - the content of the message

Returns:

an Object indicating the result of the incomingRequest. If there was a failure a Failure object is

returned

Overrides:

incomingRequest in class MessageUser

public boolean suspend()

This method overrides the method in MessageUser and suspends it

Overrides:

suspend in class MessageUser

public boolean resume()

This method overrides the method in MessageUser and resumes it

Overrides:

resume in class MessageUser

public boolean isSuspended()

This method returns whether AcMsgUser is suspended or not

Overrides:

isSuspended in class MessageUser

public boolean isReceiverPresent(byte att)

This method overrides the LocatorUser method and checks that if the message is for agent then is it present

Parameters:

att - the attribute of the message

Returns:

whether agent is present or not

Overrides:

isReceiverPresent in class MessageUser

protected ClassLoader getClassLoader(Message m)

This method overrides the MessageUser method to return the JarClassLoader of the Bucket in case message is moa application specific

Parameters:

m - the Message

Returns:

the ClassLoader for the message

Overrides:

getClassLoader in class MessageUser

8.4 AcTakedownMgr

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
          |
          +----ORG.opengroup.moa.ac.AcTakedownMgr
```

public class AcTakedownMgr extends EnvironmentChild implements AgentStateListener, Runnable

Every agent environment has one AcTakedownMgr which manages the takedown of agents on that MOA node, once their respective lifetime expires. The lifetime taken into consideration is the local lifetime of the agent at this node. It does so, by maintaining a sorted Vector list of agents according to the time at which they have to be taken down.

The agent which has to be taken down at the earliest is at top of the list. When any agent arrives at a node, according to its local lifetime, it is inserted at appropriate position in the list. This class starts a thread which waits for the duration of remaining lifetime of the agent at the top of the list. After that duration it wakes up and takes down that agent. Whenever a new agent comes on the top of agent Vector list, this thread is notified to wait for a different duration.

8.4.1 Constructor Index

AcTakedownMgr()

8.4.2 Method Index

addAC(AC, long) Adds AC according to their end of Life.

agentStateChange(AgentStateEvent) This method is called by the Mover whenever there is a change in the state of the agent.

init() method called during wiring of this component into parent environment.

main(String[ ])

removeAC(AC) This thread removes AC from the list

run() This method overrides the run() method inplements the run() method of java.lang.Runnable interface.

8.4.3 Constructors

AcTakedownMgr public AcTakedownMgr()

8.4.4 Methods

public void init()

method called during wiring of this component into parent environment. It initializes the variables, adds itself to the list of agentStateListener in Mover, and starts the takedown thread,

Overrides:

init in class EnvironmentChild

public synchronized void run()

This method overrides the run() method implements the run() method of java.lang.Runnable interface. It waits for the duration of remaining lifetime of the agent at the top of the list. After that duration it wakes up and takes down that agent. Whenever a new agent comes on the top of agent Vector list, this thread is notified to wait for a different duration.

public synchronized void addAC(AC ac, long lifetime)

Adds AC according to their end of Life. The AC which has to be taken down earliest is inserted at the top. Whenever a new AC is inserted at the top of list, the takedown thread is notified to now wait for a different duration of time.

Parameters:

ac - the AC to be inserted

lifetime - the local lifetime of the agent at this node.

public synchronized void removeAC(AC ac)

This thread removes AC from the list

Parameters:

ac - the AC to be removed from the list

public void agentStateChange(AgentStateEvent ase)

This method is called by the Mover whenever there is a change in the state of the agent. Based on whether it is an AgentArivalEvent or an AgentDepartureEvent AC is either added to the acList, or removed from it.

Parameters:

ase - the type of AgentStateEvent which prompted the method being called

public static void main(String arg[])

8.5 AgentChannelAccess

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.AgentChannelAccess
```

public class AgentChannelAccess extends Object implements Serializable

It is an AC ChannelAccess and is constructed either in openChannel method of AcChannelMgr or in the processRequest method

8.5.1 Constructor Index

AgentChannelAccess() public Constructor for Serialization/Deserialization purpose

8.5.2 Method Index

close() Called by application to close the channel

flush() Called by application to flush the channel

getDestinationName() Returns the destination to which the channel is connected

toString() Returns the Destination to which this channel is connected

write(Object) Called by application to write to the channel

8.5.3 Constructors

AgentChannelAccess public AgentChannelAccess() public Constructor for Serialization/Deserialization purpose

8.5.4 Methods

public DestinationName getDestinationName()

Returns the destination to which the channel is connected

public String toString()

Returns the Destination to which this channel is connected

Overrides:

toString in class Object

public synchronized void close() throws InvalidStateException

Called by application to close the channel

public synchronized void flush() throws InvalidStateException, IOException

Called by application to flush the channel

public synchronized void write(Object data) throws InvalidStateException, IOException

Called by application to write to the channel

Parameters:

data - the Object to be written

8.6 AgentIncomingChannel

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.AgentIncomingChannel
```

public final class AgentIncomingChannel extends Object implements AgentChannelRequest

Constructed by the AcChannelMgr in the processRequest method when an IncomingChannel object is received in the SandboxRequest item. It is passed to the Place as an item in the receiveIncomingChannel method Once a decision has been made to accept the channel or not, the response is sent via provided accept/reject methods.

8.6.1 Constructor Index

AgentIncomingChannel(DestinationName, IncomingChannel, AcChannelMgr)

Constructor called by the AcChannelMgr

8.6.2 Method Index

accept(Api) This method is called by an Api(agent/place) to accept the channel request.

getOriginatorName() Returns the originator of this channel

openToRequest(AgentChannelAccess) This method is called to open a channel given an AgentChannelAccess It is called by either the AcChannelMgr to open the channel without the application being made aware of it or by the accept method of this class

reject(Failure) Rejects the incoming channel.

8.6.3 Constructors

AgentIncomingChannel public AgentIncomingChannel(DestinationName dName, IncomingChannel ic, AcChannelMgr acm)

Constructor called by the AcChannelMgr

Parameters:

dName - originator of the IncomingChannel

ic - the IncomingChannel object received by the AcChannelMgr

acm - the AcChannelMgr which calls this constructor

8.6.4 Methods

public AgentChannelAccess accept(Api source)

This method is called by an Api(agent/place) to accept the channel request. It calls the openToRequest method of this class.

Parameters:

source - the source (agent/Place) accepting this channel

Returns:

the AgentChannelAccess constructed as a result of accepting the channel

public void reject(Failure reason)

Rejects the incoming channel.

public DestinationName getOriginatorName()

Returns the originator of this channel

public void openToRequest(AgentChannelAccess aca)

This method is called to open a channel given an AgentChannelAccess It is called by either the AcChannelMgr to open the channel without the application being made aware of it or by the accept method of this class

Parameters:

aca - the AgentChannelAccess whose ChannelAccess has to be set

8.7 ChannelClose

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.ChannelClose
```

public class ChannelClose extends Object

An instance of this object is written across a channel when an agent/place wants to close the channel to a destination. The receiver of this object acknowledges it by sending back a ChannelClosed.

8.7.1 Constructor Index

ChannelClose()

8.7.2 Constructors

ChannelClose public ChannelClose()

8.8 ChannelClosed

```
java.lang.Object
  |
  +----ORG.opengroup.moa.ac.ChannelClosed
```

public class ChannelClosed extends Object

An instance of this object is written to a channel by an agent/place/AC when it receives a ChannelClose object

8.8.1 Constructor Index

ChannelClosed()

8.8.2 Constructors

ChannelClosed ublic ChannelClosed()

8.9 ChannelHashTable

```
java.lang.Object
  |
  +----ORG.opengroup.moa.ac.ChannelHashtable
```

public class ChannelHashtable extends Object implements Serializable

8.9.1 Constructor Index

ChannelHashtable() public constructor used during Serialization process

8.9.2 Method Index

getACA(DestinationName) When a channel is open to a given destination, this method returns the AgentChannelAccess object for that channel.

getACA(String) When a channel is open to a given destination, this method returns the AgentChannelAccess object for that channel.

8.9.3 Constructors

ChannelHashtable public ChannelHashtable()

public constructor used during Serialization process

8.9.4 Methods

public synchronized AgentChannelAccess getACA(DestinationName destName)

When a channel is open to a given destination, this method returns the AgentChannelAccess object for that channel. Otherwise null is returned.

Parameters:

destName - the DestinationName of the agent/place to which this channel is connected at the other end

Returns:

an AgentChannelAccess object for the given DestinationName or null if the channel is not present

public synchronized AgentChannelAccess getACA(String destName)

When a channel is open to a given destination, this method returns the AgentChannelAccess object for that channel. Otherwise null is returned.

Parameters:

destName - the destination of the agent/place to which this channel is connected at the other end

Returns:

an AgentChannelAccess object for the given DestinationName or null if the channel is not present

8.10 ChannelMove

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.ChannelMove
```

public class ChannelMove extends Object

An instance of this class is written to a channel by an agent when it is trying to move to another location. The receiver of ChannelMove object acknowledges it by sending back a ChannelMoved object

8.10.1 Constructor Index

ChannelMove()

8.10.2 Constructors

ChannelMove public ChannelMove()

8.11 ChannelMoved

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.ChannelMoved
```

public class ChannelMoved extends Object

An instance of this object is written across a channel by an agent/place/AC to acknowledge a ChannelMove object

8.11.1 Constructor Index

ChannelMoved()

8.11.2 Constructors

ChannelMoved public ChannelMoved()

8.12 KillAgent

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.KillAgent
```

public class KillAgent extends Object

This class represents one of the commands on an agent. This is sent by an operator to kill the agent and related threads.

8.12.1 Constructor Index

KillAgent()

8.12.2 Constructors

KillAgent public KillAgent()

8.13 LogService

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.moa.ac.LogService
```

public class LogService extends EnvironmentChild

A bean which manages all local logs, as well as message routing.

8.13.1 Constructor Index

LogService()

8.13.2 Method Index

clearLocalLogs()

getLocalLogs(int)

getLocalTotal()

getLowestPriority()

log(Api, String, String) Log an item.

8.13.3 Constructors

LogService public LogService()

8.13.4 Methods

public final synchronized void clearLocalLogs()

public final Vector getLocalLogs(int priority)

public final int getLocalTotal()

public int getLowestPriority()

public final synchronized void log(Api src, String logType, String logData)

Log an item.

8.14 LogType

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.LogType
```

public final class LogType extends Object

Defines a destination and a priority for each type of log item.

8.14.1 Constructor Index

LogType()

LogType(Family, String)

8.14.2 Method Index

getAltDestinationName()

getDestinationName()

getName()

getPriority()

isOnlyLocal()

toString()

8.14.3 Constructors

LogType public LogType()

LogType public LogType(Family family, String s)

8.14.4 Methods

public String getName()

public boolean isOnlyLocal()

public DestinationName getDestinationName()

public DestinationName getAltDestinationName()

public int getPriority()

public String toString()

Overrides:

toString in class Object

8.15 OpenChannelRequest

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.OpenChannelRequest
```

public final class OpenChannelRequest extends Object implements AgentChannelRequest

Constructed by the AcMsgUser in the incomingRequest method when a request is received to open the channel It is passed to Agent as an item in the recvIncomingChannel method Once a decision has been made to accept the channel or not, the response is sent via provided accept/reject methods.

8.15.1 Constructor Index

OpenChannelRequest(String, String, int, byte)

This constructor is called from the AcChannelMgr when a request to open a channel to an agent is received

8.15.2 Method Index

accept(Api) This method is called by an agent to accept the channel request.

getOriginatorName() This method returns the DestinationName instance of the originator of request

openToRequest(AgentChannelAccess) This method is called to open a channel given an AgentChannelAccess It is called by either the AcChannelMgr to open the channel without the application being made aware of it or by the accept method of this class

reject(Failure) This method is used to reject the channel request

8.15.3 Constructors

OpenChannelRequest public OpenChannelRequest(String originatorName, String originatorHost, int originatorPort, byte att)

This constructor is called from the AcChannelMgr when a request to open a channel to an agent is received

Parameters:

originatorName - the name of the originator of the request

originatorHost - the name of the originating host

originatorPort - the name of the originating port

att - the attribute of the channel request

8.15.4 Methods

public AgentChannelAccess accept(Api source)

This method is called by an agent to accept the channel request.

Parameters:

source - the agent calling the method

Returns:

the channel created as a result of accepting the request.

public void reject(Failure reason)

This method is used to reject the channel request

Parameters:

reason - the reason the request is rejected

public DestinationName getOriginatorName()

This method returns the DestinationName instance of the originator of request

Returns:

the originator of the request

public void openToRequest(AgentChannelAccess aca)

This method is called to open a channel given an AgentChannelAccess It is called by either the AcChannelMgr to open the channel without the application being made aware of it or by the accept method of this class

Parameters:

aca - the AgentChannelAccess whose ChannelAccess has to be set

8.16 ResumeAgent

java.lang.Object

```
   +----ORG.opengroup.moa.ac.ResumeAgent
```

public class ResumeAgent extends Object

This class represents one of the commands on an agent. This is sent by an operator to resume the agent threads if they were suspended

8.16.1 Constructor Index

ResumeAgent()

8.16.2 Constructors

ResumeAgent public ResumeAgent()

8.17 SuspendAgentMvmt

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.SuspendAgentMvmt
```

public class SuspendAgentMvmt extends Object

This class represents one of the commands on an agent. This is sent by an operator to suspend the agent movement

8.17.1 Constructor Index

SuspendAgentMvmt()

8.17.2 Constructors

SuspendAgentMvmt public SuspendAgentMvmt()

8.18 SuspendAgentThread

Class ORG.opengroup.moa.ac.SuspendAgentThread

```
java.lang.Object
   |
   +----ORG.opengroup.moa.ac.SuspendAgentThread
```

public class SuspendAgentThread extends Object

This class represents one of the commands on an agent. This is sent by an operator to suspend the agent threads

8.18.1 Constructor Index

SuspendAgentThread()

8.18.2 Constructors

SuspendAgentThread public SuspendAgentThread()

9 moa.api

This package provides the interfaces that the agent developer extends to build a MOA system. The primary classes Agent and Place are extended to create the mobile part of the application.

Agent provides the functionality that is needed of the mobile application. Agents are described by AgentProperties that encapsulate an Agent's capabilities.

Place is complementary to the agent since an Agent resides in a Place in a given agent system. Agents may leave Places behind upon migration. They are also the entity where meetings are held.

Agents can avail of predefined Services and have limits set on their usage.

Meet provides agents with the facility to confer among a collection of invited agents.

9.1 Agent

```
java.lang.Object
   |
   +----ORG.opengroup.moa.api.Api
           |
           +----ORG.opengroup.moa.api.Agent
```

public class Agent extends Api implements Runnable

9.1.1 Constructor Index

Agent()

9.1.2 Method Index

application(boolean) Called by run to start the agent.

clone(int, int, Agent) Forks a child agent.

clone(int, int, Agent, AEname) Forks a child agent.

closeOpenChannels() Close all channels opened by the Agent.

getAgentChannelAccess(DestinationName) When a channel is open to a given destination, this method returns the AgentChannelAccess object for that channel.

getAgentPolicyClone() Returns a copy of the agent policy. getAlternateHomeAgentEnvironment()

getAlternateHomeAgentEnvironment Returns an alternative to the home agent environment.

getFamilyName() Returns the name of the family the agent belongs to.

getGenerationNumber() Returns the agent name extension, which denotes the generation this particular agent belongs to.

getHomeAgentEnvironment() Returns the agent environment where the agent was created.

getHomeName() Returns the name of the agent's home place.

getItinerary() Returns itinerary for this agent

getLastChildName() Returns the extension to the generation number used to create a unique name for the last child cloned.

getLaunchNumber() Returns a number associated with every instance of an agent, started by one or more users at an agent environment.

getLocalPolicyClone() Returns a copy of the local policy.

getLocationStrategy() Returns the Location Strategy for this Agent

getName() Returns the name of the agent.

getOwner() Returns a system dependent variable, only interpreted at the home node--not a basis for security.

getPlace() Returns the agent's place.

getService(String) Locate a local Service.

getStartTime() Returns the time when the agent started at the current agent environment.

hasAltHome() Returns true when there is an alt-home.

initialize(Place, Policy, Policy, AgentProperties, AC, boolean) Called by AC when an Agent arrives.

log(String, String) Log an item.

meet(AgentName[], PlaceName, long, long) Set up a meeting with other agents.

move(AEname, boolean) Move to another jvm, optionally leaving the place behind.

openChannel(DestinationName) Open a channel from the Agent to an Agent, Place, or Server.

println(String) Prints a line prefaced with a timestamp and the agent name.

run() Called by initialize to start the agent.

sendMessage(DestinationName, Object) Send a 2-way message from the Agent to an Agent, Place, or Server.

snapShot() flush all persistent agent data to the home place

takedownAC(Failure) Terminate the Agent and Place.

9.1.3 Constructors

Agent public Agent()

9.1.4 Methods

public final void initialize(Place place, Policy localPolicy, Policy agentPolicy, AgentProperties properties, AC ac, boolean returning)

Called by AC when an Agent arrives.

public final void run()

Called by initialize to start the agent.

public void application(boolean returning)

Called by run to start the agent. Generally overridden by the application.

public final AgentChannelAccess getAgentChannelAccess(DestinationName destNam) throws InvalidStateException

When a channel is open to a given destination, this method returns the AgentChannelAccess object for that channel. Otherwise null is returned.

public final AgentChannelAccess openChannel(DestinationName destName) throws IOException, InvalidStateException

Open a channel from the Agent to an Agent, Place, or Server.

public final void closeOpenChannels() throws InvalidStateException

Close all channels opened by the Agent.

public final Object sendMessage(DestinationName destName, Object content) throws InvalidStateException

Send a 2-way message from the Agent to an Agent, Place, or Server.

public final void log(String logType, String logData) throws InvalidStateException

Log an item.

public final void takedownAC(Failure reason)

Terminate the Agent and Place.

public final Service getService(String fullPathName) throws InvalidStateException

Locate a local Service.

public final void move(AEname where, boolean leavePlace) throws IOException, InvalidStateException

Move to another jvm, optionally leaving the place behind.

public final void snapShot() throws IOException, InvalidStateException

flush all persistent agent data to the home place

public final Object meet(AgentName invitees[], PlaceName loc, long when, long howLong) throws InvalidStateException

Set up a meeting with other agents.

public void println(String x)

Prints a line prefaced with a timestamp and the agent name.

public final AgentName getName()

Returns the name of the agent.

public final String getOwner()

Returns a system dependent variable, only interpreted at the home node--not a basis for security.

public final String getHomeName()

Returns the name of the agent's home place.

public final String getFamilyName()

Returns the name of the family the agent belongs to.

public final int getLaunchNumber()

Returns a number associated with every instance of an agent, started by one or more users at an agent environment.

E.G. 1 is for agent-1.

public final String getGenerationNumber()

Returns the agent name extension, which denotes the generation this particular agent belongs to. E.G. ".1.2.1.2.4" indicates that the agent is the fourth child of the second child of the first child of the second child of the first instance of an agent.

(This string is initially empty. When a clone is made, "." + the child number is appended to the parent's generation number to form the clone's generation number.)

public final int getLastChildName()

Returns the extension to the generation number used to create a unique name for the last child cloned.

public final AEname getHomeAgentEnvironment()

Returns the agent environment where the agent was created.

public final AEname getAlternateHomeAgentEnvironment()

Returns an alternative to the home agent environment.

public final int getLocationStrategy()

Returns the Location Strategy for this Agent

public final AEname[] getItinerary()

Returns itinerary for this agent

public final long getStartTime()

Returns the time when the agent started at the current agent environment. If setStart-Time was never called, 0 is returned.

public final boolean hasAltHome()

Returns true when there is an alt-home.

public final Policy getLocalPolicyClone() throws CloneNotSupportedException

Returns a copy of the local policy.

public final Policy getAgentPolicyClone() throws CloneNotSupportedException

Returns a copy of the agent policy.

public final Place getPlace()

Returns the agent's place. WARNING: All references to place should be transient.

protected final AgentName clone(int noOfClones, int noOfPlaces, Agent child, AEname destHost) throws CloneNotSupportedException, InvalidStateException

Forks a child agent.

protected final AgentName clone(int noOfClones, int noOfPlaces, Agent child) throws CloneNotSupportedException, InvalidStateException

Forks a child agent.

9.2 AgentProperties

```
java.lang.Object
   |
   +----ORG.opengroup.moa.api.AgentProperties
```

public final class AgentProperties extends Object implements Serializable, Cloneable

This serializable class holds the various properties and resource limits of an agent which an agent may not directly modify.

The naming convention used here is described in the Java Beans 1.0 API specification, section 8.3: Design Patterns for Properties.

As a proper bean, given a visual bean development package, objects of this class should be fully configurable without further programming. (Assuming the availability of a URLEditor class.)

9.2.1 Constructor Index

AgentProperties()

9.2.2 Method Index

clone() cloneAgentProperties(int, int)

Clone AgentProperties, allocating clones and places to the clone while reducing the clones and places of the parent.

getAgentName() Returns the full name of the agent.

getAlternateHomeAgentEnvironment() Returns an alternative to the home agent environment.

getEagerChannelOpen() getFamilyName()

Returns the name of the family the agent belongs to.

getFamilyNumber() Returns a number associated with every instance of an agent, started by one or more users at an agent environment.

getGenerationNumber() Returns the agent name extension, which denotes the generation this particular agent belongs to.

getHomeAgentEnvironment() Returns the agent environment where the agent was created.

getHomeLife() Return the length of time, in milliseconds, snapshots to be retained by HomeService.

getHomeName() Returns the name of the agent's home place.

getItinerary() Return the itinerary used by nameServer

getItineraryString() Return the itinerary used by nameServer

getLastChildName() Returns the extension to the generation number used to create a unique name for the last child cloned.

getLocationStrategy() Return the locationStrategy used by nameServer

getMaxSnapshots() Return the maximum number of snapshots to be retained by HomeService.

getOwner() Returns a system dependent variable, only interpreted at the home node--not a basis for security.

getStartTime() Returns the time when the agent started at the current agent environment.

hasAltHome() setAlternateHomeAgentEnvironment(URL)

Sets an alternative to the home agent environment.

setEagerChannelOpen(boolean)

setFamilyName(String) Sets the name of the family the agent belongs to.

setFamilyNumber(int) Sets a number associated with every instance of an agent, started by one or more users at an agent environment.

setHomeAgentEnvironment(URL) Sets the agent environment where the agent was created.

setHomeLife(int) Set the length of time, in milliseconds, snapshots to be retained by HomeService.

setItinerary(AEname[]) Set the Itinerary used by name server

setItineraryString(String[]) Set the Itinerary used by name server

setLocationStrategy(int) Set the locationStrategy used by name server

setMaxSnapshots(int) Set the maximum number of snapshots to be retained by HomeService.

setOwner(String) Sets a system dependent variable, only interpreted at the home node--not a basis for security.

setStartTime() Sets time when the agent started at the current agent environment.

9.2.3 Constructors

AgentProperties public AgentProperties()

9.2.4 Methods

public Object clone()

Overrides:

clone in class Object

public AgentProperties cloneAgentProperties(int clones, int places)

Clone AgentProperties, allocating clones and places to the clone while reducing the clones and places of the parent.

Parameters:

clones - Number of clones allocated to the clone.

places - Number of places allocated to the clone.

Returns:

The clone, or null if there were not enough clones or places.

public String getOwner()

Returns a system dependent variable, only interpreted at the home node--not a basis for security.

public void setOwner(String s)

Sets a system dependent variable, only interpreted at the home node--not a basis for security.

public String getHomeName()

Returns the name of the agent's home place.

public AgentName getAgentName()

Returns the full name of the agent.

public String getFamilyName()

Returns the name of the family the agent belongs to.

public void setFamilyName(String s)

Sets the name of the family the agent belongs to.

public int getFamilyNumber()

Returns a number associated with every instance of an agent, started by one or more users at an agent environment.

E.G. 1 is for agent-1.

public void setFamilyNumber(int i)

Sets a number associated with every instance of an agent, started by one or more users at an agent environment. E.G. 1 is for agent-1.

public String getGenerationNumber()

Returns the agent name extension, which denotes the generation this particular agent belongs to. E.G. ".1.2.1.2.4" indicates that the agent is the fourth child of the second child of the first child of the second child of the first instance of an agent. (This string is initially empty. When a clone is made, "." + the child number is appended to the parent's generation number to form the clone's generation number.)

public int getLastChildName()

Returns the extension to the generation number used to create a unique name for the last child cloned.

public URL getHomeAgentEnvironment()

Returns the agent environment where the agent was created.

public void setHomeAgentEnvironment(URL u)

Sets the agent environment where the agent was created.

public URL getAlternateHomeAgentEnvironment()

Returns an alternative to the home agent environment.

public void setAlternateHomeAgentEnvironment(URL u)

Sets an alternative to the home agent environment.

public long getStartTime()

Returns the time when the agent started at the current agent environment.

If setStartTime was never called, 0 is returned.

public void setStartTime()

Sets time when the agent started at the current agent environment. This method can only be called once when an AgentProperties object is unserialized.

public boolean hasAltHome()

public boolean getEagerChannelOpen()

public void setEagerChannelOpen(boolean t)

public int getMaxSnapshots()

Return the maximum number of snapshots to be retained by HomeService.

public void setMaxSnapshots(int i)

Set the maximum number of snapshots to be retained by HomeService.

public int getHomeLife()

Return the length of time, in milliseconds, snapshots to be retained by HomeService.

public void setHomeLife(int i)

Set the length of time, in milliseconds, snapshots to be retained by HomeService.

public String[] getItineraryString()

Return the itinerary used by nameServer

public void setItineraryString(String itinString[])

Set the Itinerary used by name server

public AEname[] getItinerary()

Return the itinerary used by nameServer

public void setItinerary(AEname itinerary[])

Set the Itinerary used by name server

public int getLocationStrategy()

Return the locationStrategy used by nameServer

public void setLocationStrategy(int locationStrategy)

9.3 Api

```
java.lang.Object
   |
   +----ORG.opengroup.moa.api.Api
```

public class Api extends Object implements Serializable

This interface provides methods which should be implemented by both Agent and Place.

9.3.1 Constructor Index

Api()

9.3.2 Method Index

data(AgentChannelAccess, Object) Called by Ac.

getChannelHashtable() Called by Ac.

meetOffer(AgentName, Meet) Called by Ac, Accept reject meet offer from another agent, based on the meet object (generally overriden byapplication).

receiveIncomingChannel(AgentChannelRequest) Called by Ac.

receiveMessage(DestinationName, Object) Called by Ac.

9.3.3 Constructors

Api public Api()

9.3.4 Methods

public final ChannelHashtable getChannelHashtable()

Called by Ac.

public void receiveIncomingChannel(AgentChannelRequest acr)

Called by Ac. Generally overridden by the application.

public void data(AgentChannelAccess agentChannelAccess, Object content)

Called by Ac. Generally overridden by the application.

public Object receiveMessage(DestinationName origName, Object content)

Called by Ac. Generally overridden by the application.

public final Object meetOffer(AgentName offerer, Meet meetObj)

Called by Ac, Accept reject meet offer from another agent, based on the meet object (generally overridden by application).

9.4 Meet

```
java.lang.Object
  |
  +----ORG.opengroup.moa.api.Meet
```

public class Meet extends Object implements Serializable

9.4.1 Constructor Index

Meet(AgentName[], PlaceName, long)

9.4.2 Method Index

meetInProgress()

meetSucceeded()

9.4.3 Constructors

public Meet(AgentName invitees[], PlaceName loc,long when)

9.4.4 Methods

public boolean meetInProgress()

public boolean meetSucceeded()

9.5 Place

```
java.lang.Object
  |
  +----ORG.opengroup.moa.api.Api
        |
        +----ORG.opengroup.moa.api.Place
```

public class Place extends Api

9.5.1 Constructor Index

Place()

9.5.2 Method Index

closeOpenChannels() Close all channels opened by the Place.

getAgentChannelAccess(DestinationName) When a channel is open to a given destination, this method returns the AgentChannelAccess object for that channel.

getName() Returns the name of the place.

getService(String) Locate a local Service.

initialize(AC, PlaceName) Called by AC.

log(String, String) Log an item.

openChannel(DestinationName) Open a channel from the Place to an Agent, Place, or Server.

println(String) Prints a line prefaced with a timestamp and the agent name.

sendMessage(DestinationName, Object) Send a 2-way message from the Place to an Agent, Place, or Server.

takedownAC(Failure) Terminate the Agent and Place.

9.5.3 Constructors

Place public Place()

9.5.4 Methods

public final void initialize(AC ac,mPlaceName name)

Called by AC.

public final AgentChannelAccess getAgentChannelAccess(DestinationName destNam) throws InvalidStateException

When a channel is open to a given destination, this method returns the AgentChannelAccess object for that channel. Otherwise null is returned.

public final AgentChannelAccess openChannel(DestinationName destName) throws IOException, InvalidStateException

public final void closeOpenChannels() throws InvalidStateException

Close all channels opened by the Place.

public final Object sendMessage(DestinationName destName, Object content) throws InvalidStateException

Send a 2-way message from the Place to an Agent, Place, or Server.

public final void log(String logType, String logData) throws InvalidStateException

Log an item.

public final void takedownAC(Failure reason)

Terminate the Agent and Place.

public final Service getService(String fullPathName) throws InvalidStateException

Locate a local Service.

public void println(String x)

Prints a line prefaced with a timestamp and the agent name.

public final PlaceName getName()

Returns the name of the place.

9.6 Service

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.moa.api.Service
```

public class Service extends EnvironmentChild

Identifies a service that is directly accessible to Agent and Place.

9.6.1 Constructor Index

Service()

9.6.2 Method Index

getService(String) Returns the reference to be passed to the application.

getServiceDescription() Returns a multi-line description of the service.

setServiceDescription(String) Can only be set at design time

9.6.3 Constructors

Service

public Service()

9.6.4 Methods

public Object getService(String requestingAgent)

Returns the reference to be passed to the application.

public String getServiceDescription()

Returns a multi-line description of the service.

public void setServiceDescription(String description)

Can only be set at design time

Parameters:

description - The description of the service. (Can be a multiline comment)

10 moa.logger

The ORG.opengroup.moa.logger package provides a class that allows messages to be logged to persistent media. The Logger is the object that does the logging of LogItems. The Logger responds to LogQueries based on available data.

10.1 LogItem

```
java.lang.Object
   |
   +----ORG.opengroup.moa.logger.LogItem
```

public class LogItem extends Object implements Serializable

This class defines a log message. These are the types of messages that an agent generates. Once created, a log message cannot be modified. This class has the following members:

the time when this object was created

the host computer related to this item

the port on the host computer

the agentName is the agent/component requesting the log

the activity is the kind of activity being logged

the message is the message string to be logged

An ID number is provided that allows the user to group collections of LogItems together. This number may be manipulated at will by the user.

10.1.1 Field Index

activity Name of the activity being logged

agentName Name of the agent from which the LogItem originated

host Name of the host computer from which the LogItem originated

ID An identifier that allows sets of log items to be identified as a collection.

message The message associated with this LogItem

port Number of the port of the host from which the LogItem originated

time The time this msg was recorded

10.1.2 Constructor Index

LogItem() The default constructor.

LogItem(LogItem) Creates a new LogItem from an existing one.

LogItem(long, LogItem) Creates a new LogItem from an existing one.

LogItem(long, String, int, String, String, String) Can be used to create an item with a specified timestamp

LogItem(String) Creates a LogItem in which only the message contains any useful information.

LogItem(String, int, String, String, String) Used by an agent/component when generating a log The time is set to the current system time

LogItem(StringTokenizer) Can be used to create a LogItem from a valid string representation of the item.

10.1.3 Method Index

toString() Returns a string representation of the LogItem The elements are in the following order:

ID, time, host, port, agentName, activity, message

For example:

{LogItem(12, 24234234344, host, port, agentName, activity, Message string)}

toString(boolean, String) Returns a string representation of the LogItem with the specified field separator.

10.1.4 Fields

time public final long time

The time this msg was recorded

host public final String host

Name of the host computer from which the LogItem originated

port public final int port

Number of the port of the host from which the LogItem originated

agentName public final String agentName

Name of the agent from which the LogItem originated

activity public final String activity

Name of the activity being logged

message public final String message

The message associated with this LogItem

ID public long ID

An identifier that allows sets of log items to be identified as a collection. The id is set by the user and can be manipulated as needed.

10.1.5 Constructors

LogItem public LogItem(String host, int port, String agent, String activity, String msg)

Used by an agent/component when generating a log The time is set to the current system time

Parameters:

host - the name of the computer

port - the port

LogItem public LogItem(long timestamp, String host, int port, String agent, String activity, String msg)

Can be used to create an item with a specified timestamp

LogItem public LogItem(StringTokenizer s)

Can be used to create a LogItem from a valid string representation of the item. For this purpose the string is passed as a StringTokenizer object which the user can instruct to use a specific field delimiter Beware that it is the user's responsibility to provide a valid string representation of a LogItem.

LogItem public LogItem(long timestamp, LogItem l)

Creates a new LogItem from an existing one. All elements are the same except for the timeStamp which is supplied.

LogItem public LogItem(LogItem l)

Creates a new LogItem from an existing one. All elements are the same.

LogItem public LogItem(String s)

Creates a LogItem in which only the message contains any useful information. This information is specified by the creator of this item

LogItem public LogItem()

The default constructor. This method creates a new LogItem with the time specified as current system time. No other fields contain any useful data.

10.1.6 Methods

public final String toString()

Returns a string representation of the LogItem The elements are in the following order:

ID, time, host, port, agentName, activity, message

For example:

{LogItem(12, 24234234344, host, port, agentName, activity, Message string)}

Overrides:

toString in class Object

public final String toString(boolean showID, String fieldSeparator)

Returns a string representation of the LogItem with the specified field separator.

Parameters:

showID - appends the ID number as the first element in the string

fieldSeparator - the string to separate the items in the representation

The elements are in the following order:

ID(optional), time, host, port, agentName, activity, message

For example if the field separator were to be "|" then we would see

24234234344|host|port|agentName|activity|Message goes here

if the showID was false and

12|24234234344|host|port|agentName|activity|Message goes here

if showID were true

fieldSeparator - the string to ise to separate the fields

10.2 LogQuery

```
java.lang.Object
   |
   +----ORG.opengroup.moa.logger.LogQuery
```

public final class LogQuery extends Object implements Serializable

An instance of this class is passed to the Logger when making a query.

startTime is the time from when the logs will be examined

duration is the length of time for which logs are examined

pattern is the regular expression that will be used to match desired records in the log

10.2.1 Field Index

duration

pattern

startTime

10.2.2 Constructor Index

LogQuery(long, long, String)

10.2.3 Method Index

toString()

10.2.4 Fields

startTime public final long startTime

duration public final long duration

pattern public final String pattern

10.2.5 Constructors

LogQuery public LogQuery(long startTime, long duration, String pattern)

10.2.6 Methods

public final String toString()

Overrides:

toString in class Object

10.3 Logger

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
          |
          +----ORG.opengroup.sandbox.message.MessageUser
                  |
                  +----ORG.opengroup.moa.logger.Logger
```

public class Logger extends MessageUser

This bean (module) will perform the logging of events/requests of an agent system to persistent media. This bean is highly customizable and allows the user to set the following properties:

granularity is the amount of time (in milliseconds) over which logging should take place

retentionPeriod is the amount of time (in milliseconds) for which a log will be maintained

sendRequestTimeout is the timeout value in milliseconds for the sendRequest call

logPath is the location where the logfiles should be stored

fieldSeparator is the string to be used to separate the fields in a log record when saved to disk

filePrefix is the identifier to be prefixed to the file number in the file name

ileSuffix is the identifier to be suffixed to the file number in the file name

10.3.1 Field Index

ACK Used to acknowledge a response

DONE Used to mark the end of transmission of query results

INCOMPLETE_DATA_WARNING Used to indicate that data is incomplete

INVALID_REQUEST Used to notify that request is invalid

NOTDEFINED Default value for undefined Strings

QUERY_LOGS_FOR_LAST Used to query logs from the present to x milliseconds previous

10.3.2 Constructor Index

Logger()

10.3.3 Method Index

getFieldSeparator()

getFilePrefix()

getFileSuffix()

getFlush()

getGranularity()

getLogPath()

getRetentionPeriod()

getSendRequestTimeout()

incomingRequest(String, String, int, byte, Object) Queries come in this way.

init() Bean initialization.

log(LogItem) This method and its overloaded siblings are called to log messages.

setFieldSeparator(String) The fieldSeparator is the string that separates the fields in a log record

setFilePrefix(String) the filePrefix is the identifier to be prefixed to the file number in the file namhe

setFileSuffix(String) the fileSuffix is the identifier to be suffixed to the file number in the file name

setFlush(boolean) the flush indicates whether the log record will be written to disk immediately or not.

setGranularity(long) the granularity is the amount of time in ms over which logging should take place

se**tLogPath(String)** the logPath is the location where the logfiles should be stored

setRetentionPeriod(long) the retentionPeriod is the amount of time (in milliseconds) for which a log will be maintained

setSendRequestTimeout(long)

shutdown() Used for a clean shutdown of the Logger

10.3.4 Fields

public static final String **NOTDEFINED**

Default value for undefined Strings

public static final String **INCOMPLETE_DATA_WARNING**

Used to indicate that data is incomplete

public static final String **DONE**

Used to mark the end of transmission of query results

public static final String **ACK**

Used to acknowledge a response

public static final String **INVALID_REQUEST**

Used to notify that request is invalid

public static final long **QUERY_LOGS_FOR_LAST**

Used to query logs from the present to x milliseconds previous

10.3.5 Constructor

Logger public Logger()

10.3.6 Methods

public void init()

Bean initialization. Wires the Logger bean to work with the rest of the components in the environment

Overrides:

init in class MessageUser

public Object incomingRequest(String origName,String origHost, int origPort, byte att,Object content)

Queries come in this way. When a query is made of the logger, the message is received through this method. Currently only queries are accepted. Thus messages coming in should have att set to MoaAttcode.LogQuery and content of type LogQuery

Parameters:

origName - The name of the originating agent

origHost - The name of the host computer of the agent

origPort - The port number on the originating host

att - The MoaAttcode used to identify the msg

content - The LoagQuery object

Returns:

A LogItem with the Logger.ACK in the msg field or a Failure with the message Logger.INVALID_REQUEST

Overrides:

incomingRequest in class MessageUser

See Also:

LogQuery, LogQuery

public void log(LogItem msg)

This method and its overloaded siblings are called to log messages.

Parameters:

msg - The msg to be logged

public void shutdown()

Used for a clean shutdown of the Logger

public void setGranularity(long l)

the granularity is the amount of time in ms over which logging should take place

public long getGranularity()

public void setRetentionPeriod(long l)

the retentionPeriod is the amount of time (in milliseconds) for which a log will be maintained

public long getRetentionPeriod()

public void setSendRequestTimeout(long l)

public long getSendRequestTimeout()

public void setLogPath(String l)

the logPath is the location where the logfiles should be stored

public String getLogPath()

public void setFilePrefix(String l)

the filePrefix is the identifier to be prefixed to the file number in the file name

public String getFilePrefix()

public void setFileSuffix(String l)

the fileSuffix is the identifier to be suffixed to the file number in the file name

public String getFileSuffix()

public void setFieldSeparator(String l)

The fieldSeparator is the string that separates the fields in a log record

public String getFieldSeparator()

public void setFlush(boolean b)

the flush indicates whether the log record will be written to disk immediately or not. It is quite useful when debugging as it does not wait for the buffer to flush before the user can see the log messages

public boolean getFlush()

11 moa.monitor

This package provides the interfaces that allows the collection of information and interaction with the user object. The monitor creates the user object when requested by the user applet. Attached to the monitor are multiple instances of user objects and Family objects.

There is a monitor object at every site (ae). The monitor collects information of interest to a user object from a list of sites. The information would consist of traffic volume, message counts, and selected logs. They are responsible for triggering watch objects held by the family object. Monitor must poll sandbox of interest for counts, and then forwards them to the requesting user object, as specified by the family object.

In this way, the Family, Monitor and Watch classes work together to provide the monitoring interface.

11.1 Family

```
java.lang.Object
  |
  +----ORG.opengroup.moa.monitor.Family
```

public final class Family extends Object implements Serializable

11.1.1 Constructor Index

Family()

11.1.2 Method Index

getAltHomeService()

getCountFrequency()

getCountList()

getCountList(int)

getCountListSize()

getCountVector()

getHomeService()

getLogType(String)

getLogTypeCount()

getLogTypeString(int)

getMaxLocalLogs()

getTrace()

getUserService()

getWatchList()

getWatchList(int)

getWatchListSize()

getWatchVector()

isTrace()

setCountFrequency(int) This int (property) specifies the delay (in milliseconds) between count messages.

setCountList(int, int) setCountList(int[])

This int[] (property) is a list of the number of counts to be monitored.

setCountListSize(int) This int (property) controls the number of counts to be monitored.

setHomes(String, AEname, AEname) home and altHome must be specified before getLogType can be called

setLogTypeCount(int)

setLogTypeString(int, String)

setMaxLocalLogs(int)

setTrace(boolean) This flag (property) turns on movement messages when an agent moves.

setWatchList(int, String)

setWatchList(Watch[]) This Watch[] is a list of the number of Watches.

setWatchListSize(int) This int (property) controls the number of watches.

11.1.3 Constructors

Family public Family()

11.1.4 Methods

public final boolean isTrace()

public final boolean getTrace()

public void setTrace(boolean v)

This flag (property) turns on movement messages when an agent moves.

public int getCountFrequency()

public void setCountFrequency(int v)

This int (property) specifies the delay (in milliseconds) between count messages.

public int getCountListSize()

public void setCountListSize(int s)

This int (property) controls the number of counts to be monitored. Undefined counts are initialized to -1.

public Vector getCountVector()

public int[] getCountList()

public void setCountList(int a[])

This int[] (property) is a list of the number of counts to be monitored.

public int getCountList(int i) throws ArrayIndexOutOfBoundsException

public void setCountList(int i, int v) throws ArrayIndexOutOfBoundsException

public Vector getWatchVector()

public int getWatchListSize()

public void setWatchListSize(int s)

This int (property) controls the number of watches.

public Watch[] getWatchList()

public void setWatchList(Watch a[])

This Watch[] is a list of the number of Watches.

public String getWatchList(int i) throws ArrayIndexOutOfBoundsException

public void setWatchList(int i, String v) throws ArrayIndexOutOfBoundsException

public final int getLogTypeCount()

public final void setLogTypeCount(int s)

public final String getLogTypeString(int i) throws ArrayIndexOutOfBoundsException

public final void setLogTypeString(int i, String s) throws ArrayIndexOutOfBoundsException

public final LogType getLogType(String name)

public final void setHomes(String user, AEname home, AEname altHome)

home and altHome must be specified before getLogType can be called

public final ServerName getUserService()

public final ServerName getHomeService()

public final ServerName getAltHomeService()

public final int getMaxLocalLogs()

public final void setMaxLocalLogs(int v)

11.2 Monitor

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.message.MessageUser
                   |
                   +----ORG.opengroup.moa.monitor.Monitor
```

public class Monitor extends MessageUser implements EnvironmentListener, AgentStateListener

11.2.1 Constructor Index

Monitor()

11.2.2 Method Index

agentStateChange(AgentStateEvent)

familyChange(AC, AgentProperties, Family, Family)

getCacheDelay()

getCacheExtent()

getSendMessageTimeout() Return the send message timeout

incomingXRequest(String, String, int, byte, Object, String) Called for incoming request messages.

init() Bean initialization

nestedStateChanged(EnvironmentEvent)

setCacheDelay(int) This property can only be set at design time.

setCacheExtent(int) This property can only be set at design time.

setSendMessageTimeout(int) Set send message timeout

11.2.3 Constructors

Monitor public Monitor()

11.2.4 Methods

public int getCacheDelay()

public void setCacheDelay(int v)

This property can only be set at design time.

public int getCacheExtent()

public void setCacheExtent(int v)

This property can only be set at design time.

public final void setSendMessageTimeout(int v)

Set send message timeout

public final int getSendMessageTimeout()

Return the send message timeout

public void init()

Bean initialization

Overrides:

init in class MessageUser

public Object incomingXRequest(String origName, String origHost, int origPort, byte att, Object content, String destName)

Called for incoming request messages. Must be overridden.

Overrides:

incomingXRequest in class MessageUser

public void agentStateChange(AgentStateEvent ase)

public void familyChange(AC ac, AgentProperties ap, Family oldFamily, Family newFamily)

public void nestedStateChanged(EnvironmentEvent ee)

11.3 Watch

```
java.lang.Object
  |
  +----ORG.opengroup.moa.monitor.Watch
```

public final class Watch extends Object implements Serializable

11.3.1 Constructor Index

Watch()

Watch(String)

11.3.2 Method Index

getEvent()

getEventType()

getHost()

getLimitNdx()

getLimitValue()

getNotice()

getPort()

getResponse()

isChannelEvent()

isKillAgent()

isLimitEvent()

isMessageEvent()

isNotifyUser()

isSuspendThreads()

setEvent(String)

setEventType(String)

setHost(String)

setLimitNdx(int)

setLimitValue(int)

setNotice(String)

setPort(int)

setResponse(String)

toString()

11.3.3 Constructors

Watch public Watch()

Watch public Watch(String v) throws IllegalArgumentException

11.3.4 Methods

public String toString()

Overrides:

toString in class Object

public void setEventType(String v) throws IllegalArgumentException

public String getEventType()

public void setEvent(String v) throws IllegalArgumentException

public String getEvent()

public String getHost()

public void setHost(String v)

public int getPort()

public void setPort(int v)

public int getLimitNdx()

public void setLimitNdx(int v)

public int getLimitValue()

public void setLimitValue(int v)

public void setResponse(String v) throws IllegalArgumentException

public String getResponse()

public boolean isChannelEvent()

public boolean isMessageEvent()

public boolean isLimitEvent()

public boolean isNotifyUser()

public boolean isKillAgent()

public boolean isSuspendThreads()

public String getNotice()

public void setNotice(String v)

12 moa.mover

This package provides the mechanism by virtue of which the agents can move from node to node. It provides interfaces for dealing with arrivals, departures, terminations and the logic required to perform the move.

The mover is the only facility which instantiates an AC. The launching of an agent is done by having the monitor ask the mover to do the launch. Note that this allows agents which can not move to be first instantiated their final destination. It is responsible for extracting the family from new agents and passing them to the monitor.

The mover handles the movement of agents and snapshots. It makes use of the negotiator to determine if an agent is allowed to run on that particular AE. Snapshots are handed off to the home services object. An agent/snapshot is always passed to mover as two buckets, the first containing those objects which are not to be manipulated directly by the agent. This first bucket would contain the family (debug info) object as well as agent properties and policies.

The mover is also involved in creating clones. This allows clones to be created on the site where

12.1 AgentArivalEvent

```
java.lang.Object
  |
  +----java.util.EventObject
          |
          +----ORG.opengroup.moa.mover.AgentStateEvent
                  |
                  +----ORG.opengroup.moa.mover.AgentArivalEvent
```

public class AgentArivalEvent extends AgentStateEvent

Signals that an agent has arrived.

12.1.1 Field Index

returning

timeRemaining

12.1.2 Constructor Index

AgentArivalEvent(Object, AC, AgentProperties, Family, boolean, long)

12.1.3 Fields

returning public final boolean returning

timeRemaining public final long timeRemaining

12.1.4 Constructors

AgentArivalEvent public AgentArivalEvent(Object source, AC ac, AgentProperties ap, Family family, boolean returning, long timeRemaining)

12.2 AgentDepartureEvent

```
java.lang.Object
   |
   +----java.util.EventObject
           |
           +----ORG.opengroup.moa.mover.AgentStateEvent
                   |
                   +----ORG.opengroup.moa.mover.AgentDepartureEvent
```

public class AgentDepartureEvent extends AgentStateEvent

Signals that an agent has departed.

12.2.1 Field Index

destination

placeRemaining

timeRemaining

12.2.2 Constructor Index

AgentDepartureEvent(Object, AC, AgentProperties, Family, boolean, AEname, long)

12.2.3 Fields

placeRemaining public final boolean placeRemaining

destination public final AEname destination

timeRemaining public final long timeRemaining

12.2.4 Constructors

AgentDepartureEvent public AgentDepartureEvent(Object source, AC ac, AgentProperties ap, Family family, boolean placeRemaining, AEname destination, long timeRemaining)

12.3 AgentStateEvent

```
java.lang.Object
   |
   +----java.util.EventObject
           |
           +----ORG.opengroup.moa.mover.AgentStateEvent
```

public class AgentStateEvent extends EventObject

12.3.1 Field Index

ac

ap

family

12.3.2 Constructor Index

AgentStateEvent(Object, AC, AgentProperties, Family)

12.3.3 Fields

ap public final AgentProperties ap

family public final Family family

ac public final AC ac

12.3.4 Constructors

AgentStateEvent

public AgentStateEvent(Object source, AC ac, AgentProperties ap, Family family)

12.4 AgentTerminationEvent

```
java.lang.Object
   |
   +----java.util.EventObject
           |
           +----ORG.opengroup.moa.mover.AgentStateEvent
                   |
                   +----ORG.opengroup.moa.mover.AgentTerminationEvent
```

public class AgentTerminationEvent extends AgentStateEvent

Signals that an agent has terminated.

12.4.1 Field Index

reason

12.4.2 Constructor Index

AgentTerminationEvent(Object, AC, AgentProperties, Family, Failure)

12.4.3 Fields

reason public final Failure reason

12.4.4 Constructors

AgentTerminationEvent public AgentTerminationEvent(Object source, AC ac, AgentProperties ap, Family family, Failure reason)

12.5 Mover

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
          |
          +----ORG.opengroup.sandbox.message.MessageUser
                  |
                  +----ORG.opengroup.moa.mover.Mover
```

public class Mover extends MessageUser

12.5.1 Field Index

AC_BEAN_NAME

12.5.2 Constructor Index

Mover()

12.5.3 Method Index

addAgentStateListener(AgentStateListener)

fireAgentStateChange(AgentStateEvent)

getTimeout()

incomingRequest(String, String, int, byte, Object) Called for incoming request messages.

init() Bean initialization

move(JarFile, String, int, boolean, long)

removeAgentStateListener(AgentStateListener)

setTimeout(int)

12.5.4 Fields

public static final String **AC_BEAN_NAME**

12.5.5 Constructors

Mover public Mover()

12.5.6 Methods

public void setTimeout(int tmo)

public int getTimeout()

public void init()

Bean initialization

Overrides:

init in class MessageUser

public Object incomingRequest(String origName, String origHost, int origPort, byte att, Object content)

Called for incoming request messages.

Overrides:

incomingRequest in class MessageUser

public Failure move(JarFile jf, String host, int port, boolean placeRemaining, long timeRemaining)

public void addAgentStateListener(AgentStateListener asl)

public void removeAgentStateListener(AgentStateListener asl)

public void fireAgentStateChange(AgentStateEvent ase)

12.6 AgentStateListener (Interface)

public interface AgentStateListener extends EventListener

12.6.1 Method Index

agentStateChange(AgentStateEvent)

12.6.2 Methods

agentStateChange public abstract void agentStateChange(AgentStateEvent ase)

13 moa.nameserver

The name server tracks agent location. It maintains the local information about the agent and provides it to clients upon their request. Name server supports interfaces for registering, unregistering and looking up of agent location. Looking up can either provide full information (location object), or only the current agent location.

Location information at name server is maintained only during the life time of agent. After that it is discarded. It is used by ac and monitor to determine where an agent and its clones are currently located. It also serves as a logical messaging service, delivering messages from agents and monitor to other agents.

Name server is primarily used on the source side of sending messages by either monitor or AC. On the destination, it can be invoked if the target agent is not found. This is achieved through deadName interface. In that case, name server

attempts to locate the agent either at the local hash table, or reacts according to the location strategy. As a last resort, it sends message to the home environment name server, who should ultimately know how to locate the agent. If the name server is actually residing at the home node and it still cannot locate the agent, this is considered as a failure.

Name server resolves the logical agent name into a name/location (location typically consists of host/port pair) and invokes a messenger sendMessage method. This might require sending message to the dedicated node where agent registered at that particular name server, or to multiple nodes where agent might reside, etc. depending on the location strategy. The most preferred case is that the local name server contains the actual location of the agent. In that case, message is sent directly to the node where it is believed the agent resides.

13.1 AgentName

This class contains methods for various name conversions of the agent name. It is intended for name manipulations at the nodes where agent does not already exist. At such places it is not possible to access parts of the names embedded in agentProperties. Additionally, the AgentName objects are also used for addressing all destinations of messages, or opening channels. Therefore, this class has acquired some more meaning aside of static use of methods. The objects will be created for agents and used as parameters for communication.

Based on two different needs, we have two uses of this class, one set of methods for obtaining various fields, when we do not have AgentName object, but rather the String representing it, and another set when there is the agentName as an existing object. The former set of methods can be used within MOA system, where Strings are used, and the latter at the API level where *Names are used instead.

Class ORG.opengroup.moa.nameserver.AgentName

```
java.lang.Object
|
+----ORG.opengroup.moa.nameserver.DestinationName
        |
        +----ORG.opengroup.moa.nameserver.AgentName
```

public final class AgentName extends DestinationName

This file contains methods for various name conversions of the agent name. It is intended for name manipulations at the nodes where agent does not already exist. At such places it is not possible to access parts of the names embedded in agentProperties. Additionally, the AgentName objects are also used for addressing all destinations of messages, or opening channels. Therefore, this class has acquired some more meaning aside of static use of methods. The objects will be created for agents and used as parameters for communication.

Based on two different needs, we have two uses of this class, one set of methods for obtaining various fields, when we do not have AgentName object, but rather the String representing it, and another set when there is the agentName as an existing object. The former set of methods can be used within MOA system, where Strings are used, and the latter at the API level where *Names are used instead.

The agent name consists of the following form:

h:p#f_l.g

where:

h - stands for host (home AE) (String)

p - stands for port (home AE) (int converted to String)

f - stands for family name (String)

l - stands for launch number (int converted to String)

g - stands for generation number (of the following form 1.2.3.1.2)

13.1.1 Constructor Index

AgentName(String) Create an AgentName object from the String.

AgentName(String, int, String, int, String) Create an AgentName object from the host, port, family name.

13.1.2 Method Index

checkName() Checking (in)validity of the agent name object.

checkName(String) Checking (in)validity of the String representing agent name.

getFamily() Extract the family name from the agentName object.

getFamily(String) Extract the family name from the agentName string.

getFullPath() Return the full agentName path at any host.

getFullPath(String) Create the String representing the full agentName path at any host.

getGeneration() Extract generation name from the agentName object.

getGeneration(String) Extract generation name from the agentName String.

getHomeHost() Extract the home host from the agentName object.

getHomeHost(String) Extract the home host from the string representing agentName.

getHomePort() Extract the home port from the agentName object.

getHomePort(String) Extract the home port from the agentName string.

getLaunch() Extract the launch number from the agentName object.

getLaunch(String) Extract the launch number from the agentName string.

getLaunchName() Extract the launch name from the agentName object.

getLaunchName(String) Extract the launch name from the agentName string.

main(String[]) Test program for testing static/dynamic methods with various forms of agent names.

pathToName(String) Convert fullPath to String representing the name of agent

toString() Convert agentName object to String

13.1.3 Constructors

AgentName public AgentName(String host, int port, String family, int launch, String generation)

Create an AgentName object from the host, port, family name. launch number and generation string. The first three parameters are obligatory. It is also possible to specify launch number and generation number, but they are optional.

Parameters:

host - the home host of AE where agent (or predecessor) was created

port - the home port of the AE where agent (or predecessor) was created

family - the family name of the agent

launch - the launch number (if multiple agents with the same family name and at the same site are launched)

generation - the generation number, agent ever cloned

AgentName public AgentName(String agentName)

Create an AgentName object from the String.

Parameters:

agentName - String used for building the AgentName object

13.1.4 Methods

public static String getFullPath(String agentName)

Create the String representing the full agentName path at any host. It is used for obtaining the AC in local address mapping in each AgentSystem.

Parameters:

agentName - String representing the AgentName

Returns:

String representing the full local path of the agent Name component

public String getFullPath()

Return the full agentName path at any host.

Returns:

String representing the full local path of the agent Name component

Overrides:

getFullPath in class DestinationName

See Also:

getFullPath

public static String pathToName(String fullPath)

Convert fullPath to String representing the name of agent

Parameters:

fullPath - String representing the fullPath to agent at any AE

Returns:

String name of the agent whose fullPath was provided

public static boolean checkName(String agentName)

Checking (in)validity of the String representing agent name. ":" && "#" are obligatory in the agent name string. if any other field is zero, or less than zero, it will be returned as zero in the corresponding getHomeHost, getHomePort, getFamily methods. launch and generation are allowed to be zero/null.

Parameters:

agentName - String representing the AgentName

Returns:

boolean indicating (in)validity of the agentName, false returned if name is correct

public boolean checkName()

Checking (in)validity of the agent name object. Dynamic version of checkName()

Returns:

boolean indicating (in)validity of the agentName

See Also:

checkName

public static String getHomeHost(String agentName)

Extract the home host from the string representing agentName. This method returns string from the beginning of the name till first occurrence of ":".

Parameters:

agentName - String representing the AgentName

Returns:

String representing the agent's home node *public String getHomeHost()*

Extract the home host from the agentName object.

Returns:

String representing the agent's home node

See Also:

getHomeHost

public static int getHomePort(String agentName)

Extract the home port from the agentName string. Return substring between the first occurrence of ":" and the first occurrence of "#", converted to Integer.

Returns:

int representing the agent's home AE port number

public int getHomePort()

Extract the home port from the agentName object. Return substring between the first occurrence of ":" and the first

See Also:

getHomeHost

public static String getFamily(String agentName)

Extract the family name from the agentName string. Return substring between the first occurrence of "#" and either the first occurrence of "_" (which means there is a launch number), or the first occurrence of "." (which means there is a generate. num), or the end of the string.

Returns:

String representing Family name of the input agentName String

public String getFamily()

Extract the family name from the agentName object. Invoke static version of getFamily()

Returns:

String representing Family name of the agentName object

See Also:

getFamily

public static int getLaunch(String agentName)

Extract the launch number from the agentName string. Parse the substring between the first occurrence of "_" and either the first occurrence of "." (which means there is a generate. num), or the end of the string.

Returns:

int representing the launch number

public int getLaunch()

Extract the launch number from the agentName object. Invoke the static version of this method

Returns:

int representing the launch number

See Also:

getLaunch

public static String getLaunchName(String agentName)

Extract the launch name from the agentName string. Parse substring ending with either the first occurrence of "." (which means there is a generate num), or the end of the string.

Parameters:

agentName - String representing the agentName

Returns:

String the agent name without generation number

public String getLaunchName()

Extract the launch name from the agentName object. Invoke static version of the same method

Parameters:

agentName - String representing the agentName

Returns:

String the agent name without generation number

See Also:

getLaunchName

public static String getGeneration(String agentName)

Extract generation name from the agentName String. Parse substring starting from the first occurrence of "." and until the end of the agentName.

Parameters:

String - representing the agentName

Returns:

String representing the generation number

public String getGeneration()

Extract generation name from the agentName object. Invoke the static version of this method

Returns:

String representing the generation number

See Also:

getGeneration

public String toString()

Convert agentName object to String

Returns:

String representing the agentName of this agentName object

Overrides:

toString in class Object

public static void main(String arg[])

Test program for testing static/dynamic methods with various forms of agent names.

13.2 AEName

Class ORG.opengroup.moa.nameserver.AEname

```
java.lang.Object
  |
  +----ORG.opengroup.moa.nameserver.AEname
```

public class AEname extends Object implements Serializable

The AEname class contains static methods for various name conversions of the AE name. It is specifically intended for name manipulations at the nodes where AE does not already exist. At such places it is not possible to access parts of the names embedded in system.

In addition, since recently, we switched to using the objects for addressing all destinations of messages, or opening channels. Therefore, this class has acquired some more meaning aside of static use of methods. Now the objects will be created for agents and used as parameters for communication.

Based on two different needs, we have two uses of this class, one set of methods for obtaining various fields, when we do not have AEname object, but rather the String representing it, and another set when there is the AEname as an existing object. The former set of methods can be used within MOA system, where Strings are used, and the latter at the API level where *name objects* are used instead.

The AE name consists of the following form:

h:p

where:

h - stands for host (home AE) (String)

p - stands for port (home AE) (int converted to String)

13.2.1 Constructor Index

AEname(String) Create an AEname object from the string.

AEname(String, int) Create an AEname object from the host, port.

13.2.2 Method Index

checkName() Check the aeName String for validity.

checkName(String) Check the aeName String for validity.

getHost() Extract the host name of this aeName.

getHost(String) Extract the host name of this aeName.

getPort() Extract the int representing the port of this AE.

getPort(String) Extract the int representing the port of this AE.

main(String[]) Test program for testing static/dynamic methods that manipulate AEname

toString() Convert aeName object to string

13.2.3 Constructors

AEname public AEname(String host, int port)

Create an AEname object from the host, port. These are obligatory parameters.

Parameters:

host - the name of the host where AE resides

port - the value of the port that represents this AE

AEname public AEname(String name)

Create an AEname object from the string.

Parameters:

name - the name string from which the AE object is instantiated

13.2.4 Methods

public static boolean checkName(String aeName)

Check the aeName String for validity. Verify if ":" is present, it is obligatory.

Parameters:

String - representing the name of AE

Returns:

boolean indicating if aeName is invalid; false means OK

public boolean checkName()

Check the aeName String for validity. This method invokes static version of checkName()

Returns:

boolean indicating if aeName is invalid; false means OK

See Also:

checkName

public static String getHost(String aeName)

Extract the host name of this aeName. Parse substring from the begining of the name till first occurrence of ":".

Parameters:

aeName - string representing the name of the AE

Returns:

String representing the host of the aeName.

public String getHost()

Extract the host name of this aeName. This method invokes static version of getHost() return String representing the host part of the aeName.

See Also:

getHost

public static int getPort(String aeName)

Extract the int representing the port of this AE. Parse AE string from the first occurrence of ":" till end, return this String converted to int

Parameters:

aeName - String representing the name of the AE

Returns:

int representing the port of the AE

public int getPort()

Extract the int representing the port of this AE. This method invokes its static equivalent

Returns:

int representing the port of the aeName

See Also:

getPort

public String toString()

Convert aeName object to string

Returns:

String representing the aeName.

Overrides:

toString in class Object

public static void main(String arg[])

Test program for testing static/dynamic methods that manipulate AEname

13.3 DestinationName

```
java.lang.Object
  |
  +----ORG.opengroup.moa.nameserver.DestinationName
```

public abstract class DestinationName extends Object

All classes that support names of possible destinations for messages or opening channels (Agent, Place, Server (Service)) need to support this interface.

13.3.1 Constructor Index

DestinationName()

13.3.2 Method Index

equals(Object) Check if the object is the same DestinatioName.

getFullPath() Make conversion of the Object to String, needed for the local addressing at each agent system.

getFullPath(String) Make conversion of the input String to a form, needed for the local addressing at each agent system.

stripFullPath(String) Strip the full agentName path.

13.3.3 Constructors

DestinationName public DestinationName()

13.3.4 Methods

public abstract String getFullPath()

Make conversion of the Object to String, needed for the local addressing at each agent system. For agents it will use toString(), preceded by "/AE", for place in addition, the place part of the name will be stripped, and in case of server it will be toString(), plus any local name arrangements.

Returns:

agentName representing local name

public static String getFullPath(String name)

Make conversion of the input String to a form, needed for the local addressing at each agent system. It concatenates the given string with "/AE/" as a prefix

Returns:

agentName representing local name

public boolean equals(Object obj)

Check if the object is the same DestinatioName. Check for null, if the same class and local addressing at each agent system. For agents it will use toString(), preceded by "/AE", for place in addition, the place part of the name will be stripped, and in case of server it will be toString(), plus any local name arrangements.

Returns:

agentName representing local name

Overrides:

equals in class Object

public static String stripFullPath(String fullPath)

Strip the full agentName path. There is no dynamic version of this method.

13.4 HNSLookupTree

```
java.lang.Object
  |
  +----ORG.opengroup.moa.nameserver.HNSMessage
          |
          +----ORG.opengroup.moa.nameserver.HNSLookupTree
```

public class HNSLookupTree extends HNSMessage

HNSLookupTree class is used for an object sent across the network as a consequence of the HNSLookupTree request.

13.4.1 Constructor Index

HNSLookupTree(ServerName, AgentName) Instance a HNSLookupTree object

13.4.2 Method Index

execute(HNameServer) Execute the HNSLookupTree for an agent at a destination HNS.

toString() Return a String representation of this object.

13.4.3 Constructors

HNSLookupTree public HNSLookupTree(ServerName hName, AgentName agentName)

Instance a HNSLookupTree object

Parameters:

hName - the name of the hierarchical name server this request is sent to

agentName - the name of the agent that is being looked up across the whole tree

13.4.4 Methods

public Object execute(HNameServer HNSI)

Execute the HNSLookupTree for an agent at a destination HNS.

Parameters:

HNSI - reference to the NameServer

Returns:

Object this could be either an error or the Location

Overrides:

execute in class HNSMessage

public String toString()

Return a String representation of this object.

Returns:

String representation of this object consisting of message designation, HNS name and agent name.

Overrides:

toString in class Object

13.5 HNSMessage

```
java.lang.Object
  |
  +----ORG.opengroup.moa.nameserver.HNSMessage
```

public class HNSMessage extends Object implements Serializable

13.5.1 Constructor Index

HNSMessage(ServerName) Instance a NSRegister object

13.5.2 Method Index

execute(HNameServer) Each specific NS object will override this method

13.5.3 Constructors

HNSMessage public HNSMessage(ServerName hName)

Instance a NSRegister object

Parameters:

url - The url which will identify this nameServer.

ae - reference to agent environment associated with this name server.

cacheSize - The size of the cache where location objects are stored.

Throws: IOException

Can occur when opening the server socket.

13.5.4 Methods

<u>execute</u> public Object execute(HNameServer HNSI)

Each specific NS object will override this method

Parameters:

NSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

13.6 HNSRegister

```
java.lang.Object
   |
   +----ORG.opengroup.moa.nameserver.HNSMessage
           |
           +----ORG.opengroup.moa.nameserver.HNSRegister
```

public class HNSRegister extends HNSMessage

13.6.1 Constructor Index

HNSRegister(ServerName, ServerName) Instance a HNSRegister object

13.6.2 Method Index

execute(HNameServer) Registers an agent at a location.

toString() Constructors

HNSRegister public HNSRegister(ServerName hName, ServerName child)

Instance a HNSRegister object

13.6.3 Methods

public Object execute(HNameServer HNSI)

Registers an agent at a location.

Parameters:

HNSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

Overrides:

execute in class HNSMessage

public String toString()

Overrides:

toString in class Object

13.7 HNSReturnChildren

```
java.lang.Object
  |
  +----ORG.opengroup.moa.nameserver.HNSMessage
          |
          +----ORG.opengroup.moa.nameserver.HNSReturnChildren
```

public class HNSReturnChildren extends HNSMessage

13.7.1 Constructor index

HNSReturnChildren(ServerName) Instance a HNSReturnChildren object.

13.7.2 Method Index

execute(HNameServer) Returns children from this hName server

toString()

13.7.3 Constructors

HNSReturnChildren public HNSReturnChildren(ServerName hName)

Instance a HNSReturnChildren object

13.7.4 Methods

public Object execute(HNameServer HNSI)

Returns children from this hName server

Parameters:

HNSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

Overrides:

execute in class HNSMessage

public String toString()

Overrides:

toString in class Object

13.8 HNSReturnTree

java.lang.Object

```
  |
  +----ORG.opengroup.moa.nameserver.HNSMessage
          |
          +----ORG.opengroup.moa.nameserver.HNSReturnTree
```

public class HNSReturnTree extends HNSMessage

13.8.1 Constructor Index

HNSReturnTree(ServerName) Instance a HNSReturnTree object

13.8.2 Method Index

execute(HNameServer) Return tree of this hierarchical name server

toString()

13.8.3 Constructors

HNSReturnTree public HNSReturnTree(ServerName hName)

Instance a HNSReturnTree object

13.8.4 Methods

public Object execute(HNameServer HNSI)

Return tree of this hierarchical name server

Parameters:

HNSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

Overrides:

execute in class HNSMessage

public String toString()

Overrides:

toString in class Object

13.9 HNSUnregister

```
java.lang.Object
  |
  +----ORG.opengroup.moa.nameserver.HNSMessage
          |
          +----ORG.opengroup.moa.nameserver.HNSUnregister
```

public class HNSUnregister extends HNSMessage

13.9.1 Constructor Index

HNSUnregister(ServerName, ServerName) Instance a HNSRegister object

13.9.2 Method Index

execute(HNameServer) Registers an agent at a location.

toString()

13.9.3 Constructors

HNSUnregister public HNSUnregister(ServerName hName, ServerName child)

Instance a HNSRegister object

13.9.4 Methods

public Object execute(HNameServer HNSI)

Registers an agent at a location.

Parameters:

HNSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

Overrides:

execute in class HNSMessage

public String toString()

Overrides:

toString in class Object

13.10 HNameServer

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.message.MessageUser
                   |
                   +----ORG.opengroup.moa.nameserver.LocatorUser
                           |
                           +----ORG.opengroup.moa.nameserver.NameServer
                                   |
                                   +----ORG.opengroup.moa.nameserver.HNameServer
```

public class HNameServer extends NameServer implements HNameService

13.10.1 Constructor Index

HNameServer()

13.10.2 Method Index

init() Initialization of nameServer field

lookupTree(ServerName, AgentName) Returns the current agent location if it is registered anywhere in the sub-tree maintained by the given hNameServer.

register(ServerName, ServerName) Register a name server as a child of the give hierarchical name server.

returnChildren(ServerName) Returns the children of the hname name server.

returnTree(ServerName) Returns the whole subtree of the hname name server.

unregister(ServerName, ServerName) Unregister a name server at another hierarchical NameServer.

13.10.3 Constructors

HNameServer public HNameServer()

13.10.4 Methods

public void init()

Initialization of nameServer field

Overrides:

init in class NameServer

public Object register(ServerName hName, ServerName child)

Register a name server as a child of the give hierarchical name server.

Parameters:

ServerName - Name of hierarchical name server

ServerName - Name Server to be registered

Returns:

Object Enumeration of the tree hierarchy to the top

Throws: nameServerException

Wrong parameters, destination not HNS, ..

public void unregister(ServerName hName, ServerName child)

Unregister a name server at another hierarchical NameServer.

Parameters:

ServerName - Name of hierarchical name server

ServerName - Name Server to be unregistered

Throws: nameServerException

Wrong parameters, destination not HNS, ..

public Location lookupTree(ServerName hName, AgentName agentName)

Returns the current agent location if it is registered anywhere in the sub-tree maintained by the given hNameServer.

Parameters:

ServerName - Name of hierarchical name server

agentName - The agent that is being looked up.

Returns:

Location Current agent's location or null

Throws: nameServerException

Wrong parameters, destination not HNS, ..

public Enumeration returnChildren(ServerName hname)

Returns the children of the hname name server.

Parameters:

ServerName - Hierarchical name server whose children will be returned

Returns:

Enumeration of all children

Throws: nameServerException

Wrong parameters, destination not HNS, ..

public Object returnTree(ServerName hname)

Returns the whole subtree of the hname name server.

Parameters:

ServerName - Hierarchical name server whose children will be returned

Returns:

Object Enumeration of children or itself if not hierarchical

Throws: nameServerException

Wrong parameters, destination not HNS, ..

13.11 Location

```
java.lang.Object
  |
  +----ORG.opengroup.moa.nameserver.Location
```

public class Location extends Object implements Cachable, Serializable

13.11.1 Field Index

aeName

itinerary

strategy

timeToLive

13.11.2 Method Index

copy(Location)

getEntry() Invoked by cache when this location is searched for

lastUsed() Returns time of last use in milliseconds GMT.

release() Release the item, it is not in active use.

retain() Returns the number of milliseconds the item is to be retained after its last use.

setEntry(int) Invoked by cache when this location is added into it

toString()

13.11.3 Fields

aeName public AEname aeName

strategy public int strategy

itinerary public AEname itinerary[]

timeToLive public long timeToLive

13.11.4 Methods

public synchronized long lastUsed()

Returns time of last use in milliseconds GMT.

See Also:

currentTimeMillis

public long retain()

Returns the number of milliseconds the item is to be retained after its last use.

public void release()

Release the item, it is not in active use.

public void setEntry(int ent)

Invoked by cache when this location is added into it

public int getEntry()

Invoked by cache when this location is searched for

public String toString()

Overrides:

toString in class Object

public void copy(Location other)

13.12 LocatorUser

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.message.MessageUser
                   |
                   +----ORG.opengroup.moa.nameserver.LocatorUser
```

public class LocatorUser extends MessageUser

13.12.1 Constructor Index

LocatorUser()

13.12.2 Method Index

getNameServerBean() Return the name of the name server bean

getNameServerNetBean() Return the name of the name server net bean

init() Initialization of nameServer field

sendMessage(Message) Send a message to an agent, invoked from deadMessage.

sendMessage(String, byte, Object) Send a message to an agent, invoked by higher layers (agent, AC, ...)

sendRequest(String, byte, Object, int) Send a request to an agent, invoked by higher layers (agent, AC, ...) If the agent is not present at the destination, it will be handled by deadMessage and appropriate forwarding to the home, registering or forwarding node, or it will be searched for.

setNameServerBean(String) Set the name of the name server bean

setNameServerNetBean(String) Set the name of the name server bean

13.12.3 Constructors

LocatorUser public LocatorUser()

13.12.4 Methods

public final void setNameServerBean(String nameServerBean)

Set the name of the name server bean

public final String getNameServerBean()

Return the name of the name server bean

public final void setNameServerNetBean(String nameServerNetBean)

Set the name of the name server bean

public final String getNameServerNetBean()

Return the name of the name server net bean

public void init()

Initialization of nameServer field

Overrides:

init in class MessageUser

protected Failure sendMessage(String destName, byte att, Object content)

Send a message to an agent, invoked by higher layers (agent, AC, ...)

protected Failure sendMessage(Message m)

Send a message to an agent, invoked from deadMessage.

Overrides:

sendMessage in class MessageUser

public Object sendRequest(String destName, byte att, Object content, int tmo)

Send a request to an agent, invoked by higher layers (agent, AC, ...) If the agent is not present at the destination, it will be handled by deadMessage and appropriate forwarding to the home, registering or forwarding node, or it will be searched for.

13.13 NSLookup

```
java.lang.Object
   |
   +----ORG.opengroup.moa.nameserver.NSMessage
           |
           +----ORG.opengroup.moa.nameserver.NSLookup
```

public class NSLookup extends NSMessage

13.13.1 Constructor Index

NSLookup(String) Instance a NSLookup object

13.13.2 Method Index

execute(NameServer) Registers an agent at a location.

13.13.3 Constructors

NSLookup public NSLookup(String agentName)

Instance a NSLookup object

13.13.4 Methods

public Object execute(NameServer NSI)

Registers an agent at a location.

Parameters:

NSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

Overrides:

execute in class NSMessage

13.14 NSMessage

```
java.lang.Object
   |
   +----ORG.opengroup.moa.nameserver.NSMessage
```

public class NSMessage extends Object implements Serializable

13.14.1 Constructor index

NSMessage(String) Instance a NSRegister object

13.14.2 Method Index

execute(NameServer) Each specific NS object will override this method

13.14.3 Constructors

NSMessage public NSMessage(String agentName)

Instance a NSRegister object

Parameters:

url - The url which will identify this nameServer.

ae - reference to agent environment associated with this name server.

cacheSize - The size of the cache where location objects are stored.

Throws: IOException

Can occur when opening the server socket.

13.14.4 Methods

public Object execute(NameServer NSI)

Each specific NS object will override this method

Parameters:

NSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

13.15 NSRegister

```
java.lang.Object
   |
   +----ORG.opengroup.moa.nameserver.NSMessage
           |
           +----ORG.opengroup.moa.nameserver.NSRegister
```

public class NSRegister extends NSMessage

13.15.1 Constructor Index

NSRegister(String, AEname, int, AEname[], long) Instance a NSRegister object

13.15.2 Method Index

execute(NameServer) Registers an agent at a location.

toString()

13.15.3 Constructors

NSRegister public NSRegister(String agentName, AEname aeName, int strategy,

AEname itinerary[], long timeToLive)

Instance a NSRegister object

13.15.4 Methods

public Object execute(NameServer NSI)

Registers an agent at a location.

Parameters:

NSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

Overrides:

execute in class NSMessage

public String toString()

Overrides:

toString in class Object

13.16 NSUnregister

```
java.lang.Object
   |
   +----ORG.opengroup.moa.nameserver.NSMessage
           |
           +----ORG.opengroup.moa.nameserver.NSUnregister
```

public class NSUnregister extends NSMessage

13.16.1 Constructor Index

NSUnregister(String) Instance a NSUnregister object

13.16.2 Method Index

execute(NameServer) Registers an agent at a location.

13.16.3 Constructors

NSUnregister public NSUnregister(String agentName)

Instance a NSUnregister object

13.16.4 Methods

public Object execute(NameServer NSI)

Registers an agent at a location.

Parameters:

NSI - reference to the NameServer

Throws: NameServerException

if wrong parameters or cache full

Overrides:

execute in class NSMessage

13.17 NameServer

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.message.MessageUser
                   |
                   +----ORG.opengroup.moa.nameserver.LocatorUser
                           |
                           +----ORG.opengroup.moa.nameserver.NameServer
```

public class NameServer extends LocatorUser implements NameService, DeadMessageReceiver

13.17.1 Field Index

FORWARDING

HOME

REGISTER

SEARCHING

STATIONARY

13.17.2 Constructor Index

NameServer()

NameServer(int) This constructor is used mainly for testing purposes instantiation of the NameServer is meant to be through init(), when it is instantiated as a bean.

13.17.3 Method Index

agentStateChange(AgentStateEvent)

getCacheSize()

incomingRequest(String, String, int, byte, Object) Satisfy remote requests: register, lookup, unregister the actual request comes as an object, supported by NSRegister, NSLookup and NSUnregister classes which all extend NSMessage class.

init() Initialization of nameServer field

listAllAgents() List all agents registered at this NameServer

lookupAgent(String) Returns the current agent's location if it is registered at the invoked NameServer.

lookupAgentFamily(String) Returns the current agent's location if it is registered at the invoked NameServer, or at any NameServer (in)directly pointed by the invoked NameServer.

lookupAgentLocation(String) List all agents registered at this NameServer

noReceiver(Message) The receiver of message does not exist at this MOA system, try to locate agent and pass the message to it.

register(String, AEname, int, AEname[], long) Register an agent to be located at a location.

setCacheSize(int)

setMonitor(Monitor)

suspendedReceiver(Message) The receiver of message is suspended pass the message to the monitor.

unregister(String) Unregister an agent at a NameServer.

13.17.4 Fields

HOME public static final int HOME

REGISTER public static final int REGISTER

FORWARDING public static final int FORWARDING

SEARCHING public static final int SEARCHING

STATIONARY public static final int STATIONARY

13.17.5 Constructors

NameServer public NameServer()

NameServer public NameServer(int cacheSize)

This constructor is used mainly for testing purposes instantiation of the NameServer is meant to be through init(), when it is instantiated as a bean.

13.17.6 Methods

public void setCacheSize(int cacheSize)

public int getCacheSize()

public void setMonitor(Monitor monitor)

public void init()

Initialization of nameServer field

Overrides:

init in class LocatorUser

public Location register(String agentName, AEname aeName, int strategy, AEname itinerary[], long timeToLive)

Register an agent to be located at a location. If the owner of the agent registers again with the new location, the old one will be replaced.

Parameters:

agentName - The agent to be registered.

AEname - The location where agent is located

strategy - The strategy used for locating agents

itinerary - Itinerary for agents, if it is known

timeToLive - The agent's time to live, used for cache timeout

Returns:

Location Previously registered Location object, if any

Throws: NameServerException

Can occur if wrong parameters or cache is full.

public void unregister(String agentName)

Unregister an agent at a NameServer.

Parameters:

agentName - The agent to be registered.

public AEname lookupAgent(String agentName)

Returns the current agent's location if it is registered at the invoked NameServer.

Parameters:

agentName - The agent that is being looked up.

Returns:

AEname of the name server agent was last known to be.

public Enumeration lookupAgentFamily(String agentName)

Returns the current agent's location if it is registered at the invoked NameServer, or at any NameServer (in)directly pointed by the invoked NameServer.

Parameters:

agentName - The agent whose family is being looked up.

Returns:

Enumeration of AEnames of the name server agent was last known to be.

public Enumeration listAllAgents()

List all agents registered at this NameServer

Returns:

Enumeration of all agent names

public Location lookupAgentLocation(String agentName)

List all agents registered at this NameServer

Returns:

Enumeration of all agent names

public Object incomingRequest(String origName, String origHost, int origPort, byte att, Object content)

Satisfy remote requests: register, lookup, unregister the actual request comes as an object, supported by NSRegister, NSLookup and NSUnregister classes which all extend NSMessage class.

Overrides:

incomingRequest in class MessageUser

public void noReceiver(Message message)

The receiver of message does not exist at this MOA system, try to locate agent and pass the message to it.

public Failure suspendedReceiver(Message message)

The receiver of message is suspended pass the message to the monitor.

public void agentStateChange(AgentStateEvent ase)

13.18 PlaceName

```
java.lang.Object
   |
   +----ORG.opengroup.moa.nameserver.DestinationName
```

```
|
+----ORG.opengroup.moa.nameserver.ServiceName
        |
        +----ORG.opengroup.moa.nameserver.PlaceName
```

public class PlaceName extends ServiceName

This file contains static methods for various name conversions of the place names. It is specifically intended for the name manipulations at the nodes where place does not already exist. At such nodes it is not possible to access parts of the names embedded in agentProperties. The place name consists of the following form:

agentName%h:p where agentName as presented in AgentName class h stands for host (current AE) (String) p stands for port (current AE) (int converted to String)

13.18.1 Constructor Index

PlaceName(AgentName, String, int) Create the place name given the agent name (AgentName object), host and port.

PlaceName(String) Create the place name given the String

PlaceName(String, String, int) Create the place name given the agent name (String), host and port.

13.18.2 Method Index

checkName() Dynamic version of checkName()

checkName(String) Check for correctness of the agentName and for the next occurrences of ":".

getAgentName() Dynamic version of getAgentName()

getAgentName(String) Return the agent name given a place name.

getFullPath() Dynamic version of getFullPath()

getFullPath(String) Return the full path name of the PlaceName

getHost() Dynamic version of getHost()

getHost(String) Return the Host where this place resides

getPort() Dynamic version of getPort()

getPort(String) Return the Port of the AE where this place resides.

main(String[])

printAll(String)

printAllDynamic()

toString()

13.18.3 Constructors

PlaceName public PlaceName(String agentName, String host, int port)

Create the place name given the agent name (String), host and port. All three parameters are obligatory. This constructor is mainly for testing purposes (through static methods.

PlaceName public PlaceName(AgentName agentName, String host, int port)

Create the place name given the agent name (AgentName object), host and port. All three parameters are obligatory. This constructor is instancing objects PlaceName, used in api.

PlaceName public PlaceName(String placeName)

Create the place name given the String

13.18.4 Methods

public static String getFullPath(String placeName)

Return the full path name of the PlaceName

public String getFullPath()

Dynamic version of getFullPath()

Overrides:

getFullPath in class DestinationName

public static boolean checkName(String placeName)

Check for correctness of the agentName and for the next occurrences of ":". If port is 0, it will be recognized in getPort. If host is null it will be recognized in getHost.

public boolean checkName()

Dynamic version of checkName()

public static String getAgentName(String placeName)

Return the agent name given a place name. PlaceName string is parsed for the first occurrence of "%" and substring till this occurrence is returned.

public String getAgentName()

Dynamic version of getAgentName()

public static String getHost(String placeName)

Return the Host where this place resides

public String getHost()

Dynamic version of getHost()

Overrides:

getHost in class ServiceName

public static int getPort(String placeName)

Return the Port of the AE where this place resides.

public int getPort()

Dynamic version of getPort()

Overrides:

getPort in class ServiceName

public String toString()

Overrides:

toString in class Object

public static void printAll(String placeName)

public void printAllDynamic()

public static void main(String arg[])

13.19 ServerName

```
java.lang.Object
   |
   +----ORG.opengroup.moa.nameserver.DestinationName
           |
           +----ORG.opengroup.moa.nameserver.ServiceName
                   |
                   +----ORG.opengroup.moa.nameserver.ServerName
```

public class ServerName extends ServiceName

ServerName, a class describing a server name consisting of node, port and a String uniquely identifying theserver. It could be used dynamically or statically to parse the String. The Server name consists of the following form: h:p#n where h stands for host (String) p stands for port (int converted to String) n stands for name (String)

13.19.1 Constructor Index

ServerName()

ServerName(String) Create the server name given the String

ServerName(String, int, String) Create the server name given the host, port and server name (String).

13.19.2 Method Index

checkName() Dynamic version of checkName()

checkName(String) Check for correctness of the name and for the next occurrences of ":".

getFullPath() Dynamic version of getFullPath()

getFullPath(String) Return the full path name of the serverName

getHost() Dynamic version of getHost()

getHost(String) Return the Host where this place resides

getName() Dynamic version of getName()

getName(String) Return the serverName name given a Server name.

getPort() Dynamic version of getPort()

getPort(String) Return the Port of the AE where this server resides.

main(String[])

printAll(String)

printAllDynamic()

stripFullPath(String) Strip the full agentName path.

toString()

13.19.3 Constructors

ServerName public ServerName()

ServerName public ServerName(String host, int port, String name)

Create the server name given the host, port and server name (String). All three parameters are obligatory.

ServerName public ServerName(String serverName)

Create the server name given the String

13.19.4 Methods

public static String getFullPath(String serverName)

Return the full path name of the serverName

public String getFullPath()

Dynamic version of getFullPath()

Overrides:

getFullPath in class DestinationName

public static String stripFullPath(String serverName)

Strip the full agentName path.

public static boolean checkName(String serverName)

Check for correctness of the name and for the next occurrences of ":". If port is 0, it will be recognized in getPort. If host is null it will be recognized in getHost.

public boolean checkName()

Dynamic version of checkName()

public static String getName(String serverName)

Return the serverName name given a Server name. serverName string is parsed for the first occurrence of "#" and substring till this occurrence is returned.

public String getName()

Dynamic version of getName()

public static String getHost(String serverName)

Return the Host where this place resides

public String getHost()

Dynamic version of getHost()

Overrides:

getHost in class ServiceName

public static int getPort(String serverName)

Return the Port of the AE where this server resides.

public int getPort()

Dynamic version of getPort()

Overrides:

getPort in class ServiceName

public String toString()

Overrides:

toString in class Object

public static void printAll(String serverName)

public void printAllDynamic()

public static void main(String arg[])

13.20 ServiceName

```
java.lang.Object
   |
   +----ORG.opengroup.moa.nameserver.DestinationName
           |
           +----ORG.opengroup.moa.nameserver.ServiceName
```

public abstract class ServiceName extends DestinationName

All classes that support names of possible destinations for messages or opening channels (Agent, Place, Server (Service)) need to support this interface.

13.20.1 Constructor Index

ServiceName()

13.20.2 Method Index

getHost() Return host name of the destination.

getPort() Returns the port of the residing AE in case of place, server, or the home AE's port in case of agents.

13.20.3 Constructors

ServiceName public ServiceName()

13.20.4 Methods

public abstract String getHost()

Return host name of the destination. In case of the agent, it will be the host name o the home node. In case of place and server it is currently residing host.

Returns:

hostName The agent to be registered.

public abstract int getPort()

Returns the port of the residing AE in case of place, server, or the home AE's port in case of agents.

Returns:

hostName The agent to be registered.

13.21 NameService (Interface)

public interface NameService

All classes that provide naming services must implement this interface. Currently, these are NameServer and AgentEnvironmentImpl

13.21.1 Method Index

listAllAgents() Returns the Enumeration of all agent names currently registered at this name server.

lookupAgent(String) Returns the AEname of current agent's AE if it is registered at the invoked NameServer, or at any NameServer (in)directly pointed by the invoked NameServer.

lookupAgentFamily(String) Returns the Enumeration of AEnames of current AEs of all agent's in the family of requested agent, if they are registered at the invoked NameServer, or at any NameServer (in)directly pointed by the invoked NameServer.

lookupAgentLocation(String) Returns the current agent's location if it is registered at the invoked NameServer, *do not* extend search to other NameServer(s) (in)directly pointed by the invoked NameServer.

register(String, AEname, int, AEname[], long) Register an agent to be located at a location.

unregister(String) Unregister an agent at a NameServer.

13.21.2 Methods

public abstract Location register(String agentName, AEname aeName, int strategy, AEname itinerary[], long timeToLive)

Register an agent to be located at a location. If the owner of the agent registers again with the new location, the old one will be replaced.

Parameters:

agentName - The agent to be registered.

AEname - The AE where agent is located

strategy - The strategy used for locating agents

itinerary - Itinerary for agents, if it is known

timeToLive - The agent's time to live, used for cache timeout

Returns:

Location Previously registered location object, if any

Throws: nameServerException

Can occur if wrong parameters or cache is full.

public abstract void unregister(String agentName)

Unregister an agent at a NameServer.

Parameters:

agentName - The agent to be registered.

public abstract AEname lookupAgent(String agentName)

Returns the AEname of current agent's AE if it is registered at the invoked NameServer, or at any NameServer (in)directly pointed by the invoked NameServer.

Parameters:

agentName - The agent that is being looked up.

public abstract Enumeration lookupAgentFamily(String agentName)

Returns the Enumeration of AEnames of current AEs of all agent's in the family of requested agent, if they are registered at the invoked NameServer, or at any NameServer (in)directly pointed by the invoked NameServer.

Parameters:

agentName - The agent whose family is being looked up.

public abstract Enumeration listAllAgents()

Returns the Enumeration of all agent names currently registered at this name server.

Returns:

agentName The agent whose family is being looked up.

public abstract Location lookupAgentLocation(String agentName)

Returns the current agent's location if it is registered at the invoked NameServer, *do not* extend search to other NameServer(s) (in)directly pointed by the invoked NameServer.

Parameters:

agentName - The agent that is being looked up.

Returns:

Enumeration of all agents registered at the nameServer.

13.22 HNameService(Interface)

public interface HNameService

All classes that provide hierarchical naming services must implement this interface. Currently, this is HNameServer.

13.22.1 Method Index

lookupTree(ServerName, AgentName) Returns the current agent 'location s AE if it is registered anywhere in the sub-tree maintained by the given hNameServer.

register(ServerName, ServerName) Register a name server as a child of the give hierarchical name server.

returnChildren(ServerName) Returns the children of the hname name server.

returnTree(ServerName) Returns the whole subtree of the hname name server.

unregister(ServerName, ServerName) Unregister a name server at another hierarchical NameServer.

13.22.2 Methods

public abstract Object register(ServerName hName, ServerName child)

Register a name server as a child of the give hierarchical name server.

Parameters:

ServerName - Name of hierarchical name server

ServerName - Name Server to be registered

Returns:

Object Enumeration of tree hierarchy to the top or Failure

Throws: nameServerException

Wrong parameters, destination not HNS, ..

public abstract void unregister(ServerName hName, ServerName child)

Unregister a name server at another hierarchical NameServer.

Parameters:

ServerName - Name of hierarchical name server

ServerName - Name Server to be unregistered

Throws: nameServerException

Wrong parameters, destination not HNS, ..

public abstract Location lookupTree(ServerName hName, AgentName agentName)

Returns the current agent location s AE if it is registered anywhere in the sub-tree maintained by the given hNameServer.

Parameters:

ServerName - Name of hierarchical name server

agentName - The agent that is being looked up.

Returns:

Location Current agent's location or null

Throws: nameServerException

Wrong parameters, destination not HNS, ..

public abstract Enumeration returnChildren(ServerName hname)

Returns the children of the hname name server.

Parameters:

ServerName - Hierarchical name server whose children will be returned

Returns:

Enumeration Enumeration of all children

Throws: nameServerException

Wrong parameters, destination not HNS, ..

public abstract Object returnTree(ServerName hname)

Returns the whole subtree of the hname name server.

Parameters:

ServerName - HName server whose children will be returned

Returns:

Object Enumeration of children or itself (if not hierarchical)

Throws: nameServerException

Wrong parameters, destination not HNS, ..

14 moa.negotiate

Negotiate package contains classes executes as part of the trusted code which decides if an agent will be accepted at a particular ae or nested place (future feature). The Policy class consists of various Item objects that define certain properties of an agent or an agent system. The static class Arbitrator contains methods used for matching the properties of and agent system and an agent to verify it is acceptable for that agent system to accept the agent, and (inversely) if the agent system has all the support as required by this particular agent.

14.1 Arbitrator

java.lang.Object
  |
  +----ORG.opengroup.moa.negotiate.Arbitrator

public class Arbitrator extends Object

Arbitrator now contains a static method that calculates resulting agent's policy. The following is policy. If host has a requisite set on for an item and there is no such item supported in agent, agent's policy gets this item with memo set to "MISSING". If any of them has a value set, and it is requisite it needs to be in the range. If it is not, than is "REFUSED". Arbitrator gives Decision from two Policies. To use it Arbitrator theArbitrator = new Arbitrator(policy1, policy2); Decision theDecision = theArbitrator.getDecision();

14.1.1 Constructor Index

Arbitrator()

14.1.2 Method Index

decide(Item, Item)

getDecision(Policy, Policy)

14.1.3 Constructors

Arbitrator public Arbitrator()

14.1.4 Methods

public static Policy getDecision(Policy p1, Policy p2)

public static Item decide(Item it1, Item it2)

14.2 Item

java.lang.Object
  |
  +----ORG.opengroup.moa.negotiate.Item

14.2.1 Constructor Index

public class Item extends Object implements Serializable, Cloneable

PItem class defines a policy item. all its fields are read only by public, readable/writable by the negotiator package.

14.2.2 Constructor Index

Item()

Item(String)

Item(String, long, long, long, boolean, String)

14.2.3 Method Index

clone()

inRange(long)

main(String[])

toString()

14.2.4 Constructors

Item public Item()

Item public Item(String name, long value, long min, long max, boolean requisite, String memo)

Item public Item(String s)

14.2.5 Methdods

public String toString()

Overrides:

toString in class Object

public Object clone()

Overrides:

clone in class Object

public boolean inRange(long value)

public static void main(String arg[])

14.3 Policy

```
java.lang.Object
   |
   +----ORG.opengroup.moa.negotiate.Policy
```

public class Policy extends Object implements Serializable, Cloneable

14.3.1 Constructor Index

Policy()

Policy(Item)

Policy(String[])

14.3.2 Method Index

clone()

clone(long, long)

decrement(String)

decrement(String, long)

getItem(String)

getItemCount()

getItemString(int)

getValueOf(String)

increment(String)

increment(String, long)

isRefused()

setItemCount(int)

setItemString(int, String)

setStartTime()

setValueOf(String, long)

toString()

updateTimeRemaining() Inherited from ResourceLimits Updates the time remaining.

14.3.3 Constructors

Policy public Policy()

Policy public Policy(Item item)

Policy public Policy(String item[])

14.3.4 Methods

public Object clone() throws CloneNotSupportedException

Overrides:

clone in class Object

public Policy clone(long clones, long remainingPlaces) throws PolicyException

public void decrement(String name) throws PolicyException

public void decrement(String name, long value) throws PolicyException

public void increment(String name) throws PolicyException

public void increment(String name, long value) throws PolicyException

public long getValueOf(String name)

public void setValueOf(String name, long value)

public boolean isRefused()

public String toString()

Overrides:

toString in class Object

public final int getItemCount()

public final void setItemCount(int s)

public final String getItemString(int i) throws ArrayIndexOutOfBoundsException

public final void setItemString(int i, String s) throws ArrayIndexOutOfBoundsException

public final Item getItem(String name)

public void updateTimeRemaining() throws PolicyException

Inherited from ResourceLimits Updates the time remaining. The setTimeRemaining method should be called prior to serializing an ResourceLimits object. Start time may also be updated by this method.

public void setStartTime()

15 moa.user

The ORG.opengroup.moa.user package provides the interfaces required to interact with the agent system. It is the user entity which logs into the agent system and then launches and manages agents. This package provides the means by which to do this task.

15.1 AgentLocation

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.AgentLocation
```

public class AgentLocation extends Object implements Serializable

15.1.1 Constructor Index

AgentLocation() Constructs a AgentLocation object with the fields initialized.

AgentLocation(String, int, String) Defines an agents location

15.1.2 Method Index

getAgentName() The name of the agent whose location is defined

getHost() The host computer of this agent's location

getPort() The port of the host computer of this agent's location

15.1.3 Constructors

AgentLocation public AgentLocation()

Constructs a AgentLocation object with the fields initialized.

AgentLocation public AgentLocation(String host, int port, String agentName)

Defines an agents location

Parameters:

host - The host computer for the agent

port - The port number on the host

agentName - The name of the agent

15.1.4 Methods

public String getHost()

The host computer of this agent's location

public int getPort()

The port of the host computer of this agent's location

public String getAgentName()

The name of the agent whose location is defined

15.2 Command

```
java.lang.Object
  |
  +----ORG.opengroup.moa.user.Request
          |
          +----ORG.opengroup.moa.user.Command
```

public class Command extends Request

An agent is capable of receiving commands.This object encapsulates these commands for the agent. This object is sent to the user. It wraps around the actual command to the agent e.g. KillAgent etc.

15.2.1 Constructor Index

Command(Object, String) Initializes an object with the specified command

15.2.2 Method Index

getAgentName() The name of the agent to whom the command is being sent

getCommand() The command being issued to the agent

15.2.3 Constructors

Command public Command(Object command, String agentName)

Initializes an object with the specified command

Parameters:

command - The command object

agentname - The name of the agent for whom the command is intended

15.2.4 Methods

public Object getCommand()

The command being issued to the agent

public String getAgentName()

The name of the agent to whom the command is being sent

15.3 CreatePCA

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.Request
           |
           +----ORG.opengroup.moa.user.CreatePCA
```

public class CreatePCA extends Request

This class creates an object identifying a PreConfigured Agent. It acts as a pointer to the jar file which contains the agent.

15.3.1 Constructor Index

CreatePCA() Creates a new CreatePCA object

CreatePCA(File, JarFile) Creates a new CreatePCA object with initialized values

15.3.2 Method Index

getJar() The jar file containing the agent

getJarFilename() The file pointer to the jar file containing the agent

15.3.3 Constructors

CreatePCA public CreatePCA()

Creates a new CreatePCA object

CreatePCA public CreatePCA(File f, JarFile j)

Creates a new CreatePCA object with initialized values

Parameters:

f - the file pointer to the jar file containing the agent

j - the jar file containing the agent

15.3.4 Methods

getJarFilename public File getJarFilename()

The file pointer to the jar file containing the agent

getJar public JarFile getJar()

The jar file containing the agent

15.4 GetFamily

```
java.lang.Object
  |
  +----ORG.opengroup.moa.user.Request
          |
          +----ORG.opengroup.moa.user.GetFamily
```

public class GetFamily extends Request

Used to retrieve the Family object associated with an agent.

15.4.1 Constructor Index

GetFamily(File) Creates a new object

15.4.2 Method Index

getJarFilename() The file pointer to the agent's jar file

15.4.3 Constructors

GetFamily public GetFamily(File f)

Creates a new object

Parameters:

f - the file pointer to the jar file containing the agent

15.4.4 Methods

public File getJarFilename()

The file pointer to the agent's jar file

15.5 HomeQuery

```
java.lang.Object
  |
  +----ORG.opengroup.moa.user.Request
          |
          +----ORG.opengroup.moa.user.HomeQuery
```

public class HomeQuery extends Request

Creates a HomeQuery for the Agent

15.5.1 Constructor Index

HomeQuery(File)

15.5.2 Method Index

getJarFilename() The file pointer associated with the agent

15.5.3 Constructors

HomeQuery public HomeQuery(File f)

Parameters:

f - the file containing the agent

15.5.4 Methods

public File getJarFilename()

The file pointer associated with the agent

15.6 Launch

```
java.lang.Object
  |
  +----ORG.opengroup.moa.user.Request
          |
          +----ORG.opengroup.moa.user.Launch
```

public class Launch extends Request

The object associated with the Launch of an agent

15.6.1 Constructor Index

Launch(String, int, File) A new Launch object

15.6.2 Method Index

getHost() The host computer where the launch is taking place

getJarFilename() The file pointer to the agent's jar file

getPort() The port on the host computer where the launch is taking place

15.6.3 Constructors

Launch public Launch(String h, int p, File f)

A new Launch object

Parameters:

h - The host computer where launch is taking place

p - The port on the host computer

f - The file pointer to the jar file containing the agent

15.6.4 Methods

public File getJarFilename()

The file pointer to the agent's jar file

public String getHost()

The host computer where the launch is taking place

public int getPort()

The port on the host computer where the launch is taking place

15.7 Logon

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.Request
           |
           +----ORG.opengroup.moa.user.Logon
```

public class Logon extends Request

The object associated with a Logon to the agent system

15.7.1 Constructor Index

Logon(String, String) A new Logon object

15.7.2 Method Index

getName() The user name

getPasswd() The user's password

15.7.3 Constructors

Logon public Logon(String n,,String p)

A new Logon object

Parameters:

h - The name of the user

p - The user's password

15.7.4 Methods

public final String getName()

The user name

public final String getPasswd()

The user's password

15.8 LogonHolder

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.LogonHolder
```

public class LogonHolder extends Object implements Serializable

An object that signifies the status of a logon to the agent system

15.8.1 Constructor Index

LogonHolder() Initializes a LogonHolder object

LogonHolder(Vector) Initializes a LogonHolder object

15.8.2 Method Index

getJarFilenames() The list of file names of agents on this system

getUndeliveredMessages() The list of undelivered messages

printlh()

15.8.3 Constructors

LogonHolder public LogonHolder()

Initializes a LogonHolder object

LogonHolder public LogonHolder(Vector undelivered)

Initializes a LogonHolder object

Parameters:

undelivered - The list of undelivered messages

15.8.4 Methods

public Vector getJarFilenames()

The list of file names of agents on this system

public Vector getUndeliveredMessages()

The list of undelivered messages

public void printlh()

15.9 MeetRequest

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.Request
          |
          +----ORG.opengroup.moa.user.MeetRequest
```

public class MeetRequest extends Request

A request to meet

15.9.1 Constructor Index

MeetRequest() Initializes the request

15.9.2 Constructors

MeetRequest public MeetRequest()

Initializes the request

15.10 MoaAttcode

```
java.lang.Object
  |
  +----ORG.opengroup.moa.user.MoaAttcode
```

public class MoaAttcode extends Object

The list of attributes that signify different operations and interactions in the MOA system

15.10.1 Field Index

AgentProp

ChnlAgentToAgent

ChnlAgentToPlace

ChnlPlaceToAgent

ChnlPlaceToPlace

Command

CreatePCA

FetchSnapshot

GetFamily

HomeQuery

Launch

ListSnapshots

LogItem

Logon

LogQuery

MeetRequest

MonitorUpdate

Movement

MsgAgentToAgent

MsgAgentToPlace

MsgPlaceToAgent

MsgPlaceToPlace

Notice

Snapshot

Stats

UpdateFamily

15.10.2 Constructor Index

MoaAttcode()

15.10.3 Fields

AgentProp public static final byte AgentProp

Snapshot public static final byte Snapshot

ListSnapshots public static final byte ListSnapshots

FetchSnapshot public static final byte FetchSnapshot

Logon public static final byte Logon

Launch public static final byte Launch

GetFamily public static final byte GetFamily

UpdateFamily public static final byte UpdateFamily

Movement public static final byte Movement

Notice public static final byte Notice

Stats public static final byte Stats

MonitorUpdate public static final byte MonitorUpdate

MeetRequest public static final byte MeetRequest

CreatePCA public static final byte CreatePCA

HomeQuery public static final byte HomeQuery

LogQuery public static final byte LogQuery

LogItem public static final byte LogItem

Command public static final byte Command

MsgAgentToAgent public static final byte MsgAgentToAgent

MsgAgentToPlace public static final byte MsgAgentToPlace

MsgPlaceToAgent public static final byte MsgPlaceToAgent

MsgPlaceToPlace public static final byte MsgPlaceToPlace

ChnlAgentToAgent public static final byte ChnlAgentToAgent

ChnlAgentToPlace public static final byte ChnlAgentToPlace

ChnlPlaceToPlace public static final byte ChnlPlaceToPlace

ChnlPlaceToAgent public static final byte ChnlPlaceToAgent

15.10.4 Constructors

MoaAttcode public MoaAttcode()

15.11 MonitorUpdate

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.Request
           |
           +----ORG.opengroup.moa.user.MonitorUpdate
```

public class MonitorUpdate extends Request

A request to update the Monitor

15.11.1 Constructor Index

MonitorUpdate(String) Initializer.

15.11.2 Method Index

getJarFilename() The agent's jar file

15.11.3 Constructors

MonitorUpdate public MonitorUpdate(String f)

Initializer.

Parameters:

f - The filename for the agent's jar file

15.11.4 Methods

getJarFilename public String getJarFilename()

The agent's jar file

15.12 Movement

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.AgentLocation
           |
           +----ORG.opengroup.moa.user.Movement
```

public final class Movement extends AgentLocation implements Serializable

Defines movement primitives for an Agent

15.12.1 Field Index

ARIVED Movement action when agent arives.

DEPARTED Movement action when agent departs.

TERMINATED Movement action when agent exits or is terminated.

15.12.2 Constructor Index

Movement(byte, String, int, String)

Movement(String, int, String, Failure)

15.12.3 Method Index

getAction() The type of action enacted

getReason() Reason for the failure

toString() A string representation of this object

15.12.4 Fields

ARIVED public static final byte ARIVED

Movement action when agent arrives.

DEPARTED public static final byte DEPARTED

Movement action when agent departs.

TERMINATED public static final byte TERMINATED

Movement action when agent exits or is terminated.

15.12.5 Constructors

Movement public Movement(byte action, String host, int port, String agentName)

Parameters:

action - The type of action performed (ARRIVED, DEPARTED< TERMINATED)

host - The computer location for this agent

port - The port on the host

agentName - The name of the agent

Movement public Movement(String host, int port, String agentName, Failure reason)

Parameters:

action - The type of action performed (ARRIVED, DEPARTED< TERMINATED)

host - The computer location for this agent port - The port on the host

agentName - The name of the agent

failure - The reason the move failed

15.12.6 Methods

public byte getAction()

The type of action enacted

public Failure getReason()

Reason for the failure

public String toString()

A string representation of this object

Overrides:

toString in class Object

15.13 Notice

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.AgentLocation
           |
           +----ORG.opengroup.moa.user.Notice
```

public final class Notice extends AgentLocation

A message sent to the frontend by the monitor

15.13.1 Constructor Index

Notice(String, String, int, String, Failure)

15.13.2 Method Index

getCmdRslt() The result of the command

getWatchNotice() The watch notification string

toString() A string representation of the object

15.13.3 Constructors

Notice public Notice(String agentName, String host, int port, String watchNotice, Failure cmdRslt)

Parameters:

agentName - the agent's name

host - the host computer for the agent

port - the port on the host computer

watchNotice - a watch notification

cmdRslts - the result of the command

15.13.4 Methods

public String getWatchNotice()

The watch notification string

public Failure getCmdRslt()

The result of the command

public String toString()

A string representation of the object

Overrides:

toString in class Object

15.14 Request

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.Request
```

public class Request extends Object implements Serializable

A request sent to the user. Different types of requests extend this class. It is useless by itself.

15.14.1 Field Index

m_typecode

15.14.2 Constructor Index

Request()

15.14.3 Fields

m_typecode protected byte m_typecode

15.14.4 Constructors

Request public Request()

15.15 Snapshot

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.Request
           |
           +----ORG.opengroup.moa.user.Snapshot
```

public class Snapshot extends Request

A request for a snapshot of the agent

15.15.1 Constructor Index

Snapshot(JarFile, SnapshotID)

15.15.2 Method Index

getJar() The jar file containing the agent

getSnapshotID() The snapshot ID of the agent

15.15.3 Constructors

Snapshot public Snapshot(JarFile jar, SnapshotID id)

Parameters:

jar - the file containing the agent

id - the id of the snapshot

15.15.4 Methods

public JarFile getJar()

The jar file containing the agent

public SnapshotID getSnapshotID()

The snapshot ID of the agent

15.16 SnapshotID

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.SnapshotID
```

public class SnapshotID extends Object

The ID of the snapshot of the agent

15.16.1 Constructor Index

SnapshotID(AgentName, int)

15.16.2 Method Index

getAgentName() The agent's name

getID() The id of the snapshot desired

toString() String representation of the object

15.16.3 Constructors

SnapshotID public SnapshotID(AgentName agentName, int id)

Parameters:

agentName - the name of the agent

id - the id of the snapshot

15.16.4 Methods

public AgentName getAgentName()

The agent's name

public int getID()

The id of the snapshot desired

public String toString()

String representation of the object

Overrides:

toString in class Object

15.17 Stats

java.lang.Object

```
  |
  +----ORG.opengroup.moa.user.AgentLocation
          |
          +----ORG.opengroup.moa.user.Stats
```

public final class Stats extends AgentLocation

Encapsulation of statistics about an agent

15.17.1 Field Index

ndx

val

15.17.2 Constructor Index

Stats(String, int, String, int[], long[])

15.17.3 Method Index

toString() String representation of the object

15.17.4 Fields

ndx public final int ndx[]

val public final long val[]

15.17.5 Constructors

Stats public Stats(String host, int port, String agentName, int ndx[], long val[])

Parameters:

host - the computer the agent is on

port - the port on the host

agentName - the name of the agent

ndx - the array of indices

val - the array of values

15.17.6 Methods

public String toString()

String representation of the object

Overrides:

toString in class Object

15.18 UpdateFamily

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.Request
           |
           +----ORG.opengroup.moa.user.UpdateFamily
```

public class UpdateFamily extends Request

A request to update the Family

15.18.1 Constructor Index

UpdateFamily()

UpdateFamily(File, Object)

15.18.2 Method Index

getFamily() The family object

getJarFilename() The file containing the agent

15.18.3 Constructors

UpdateFamily public UpdateFamily()

UpdateFamily public UpdateFamily(File f, Object y)

Parameters:

f - the filename of the agent

y - the object to be updated

15.18.4 Methods

public File getJarFilename()

The file containing the agent

public Object getFamily()

The family object

15.19 User

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.message.MessageUser
                   |
                   +----ORG.opengroup.moa.nameserver.LocatorUser
                           |
                           +----ORG.opengroup.moa.user.User
```

public class User extends LocatorUser implements Serializable

Provides an interface to act with the agent system

15.19.1 Constructor Index

User()

15.19.2 Method Index

getGUIhost() The host computer for the GUI(frontend)

getGUIport() The port on the host computer for the GUI (frontend)

getHsRetryDelay() The retry delay in milliseconds

getPasswd() The user's password

getTimeOut() The time out value in milliseconds

getUserPath() The user's path

incomingRequest(String, String, int, byte, Object) Incoming messages are services here.

init() The initialization of the user object

isConnected() Connection status

setConnection(boolean)

setGUIhost(String)

setGUIport(int)

setHsRetryDelay(long)

setPasswd(String)

setTimeOut(int)

setUserPath(String)

15.19.3 Constructors

User public User()

15.19.4 Methods

public synchronized String getPasswd()

The user's password

public synchronized void setPasswd(String n)

public synchronized String getUserPath()

The user's path

public synchronized void setUserPath(String n)

public synchronized long getHsRetryDelay()

The retry delay in milliseconds

public synchronized void setHsRetryDelay(long n)

public synchronized int getTimeOut()

The time out value in milliseconds

public synchronized void setTimeOut(int n)

public synchronized boolean isConnected()

Connection status

public synchronized void setConnection(boolean b)

public synchronized String getGUIhost()

The host computer for the GUI(frontend)

public synchronized void setGUIhost(String n)

public synchronized int getGUIport()

The port on the host computer for the GUI (frontend)

public synchronized void setGUIport(int n)

public void init()

The initialization of the user object

Overrides:

init in class LocatorUser

public Object incomingRequest(String origName, String origHost, int origPort, byte att, Object content)

Incoming messages are services here.

Parameters:

origName - the name of the originating agent

origPort - the name of the originating host computer

origPort - the port on the host computer

att - the MoaAttcode signifying the type of message

content - the message

Overrides:

incomingRequest in class MessageUser

15.20 UserFetchSnapshot

```
java.lang.Object
  |
  +----ORG.opengroup.moa.user.Request
        |
        +----ORG.opengroup.moa.user.UserFetchSnapshot
```

public class UserFetchSnapshot extends Request

A request to fetch a snapshot

15.20.1 Constructor Index

UserFetchSnapshot()

UserFetchSnapshot(String, int, SnapshotID)

15.20.2 Method Index

getHost() The host from which to fetch the snapshot

getPort() The port on the host

getSnapshotID() The id of the snapshotbeing fetched

15.20.3 Constructors

UserFetchSnapshot public UserFetchSnapshot()

UserFetchSnapshot public UserFetchSnapshot(String h, int p, SnapshotID id)

Parameters:

h - The host to fetch the snapshot from

p - the port on the host

id - The id of the snapshot being fetched

15.20.4 Methods

public SnapshotID getSnapshotID()

The id of the snapshotbeing fetched

public String getHost()

The host from which to fetch the snapshot

public int getPort()

The port on the host

15.20.5 UserListSnapshots

```
java.lang.Object
  |
  +----ORG.opengroup.moa.user.Request
          |
          +----ORG.opengroup.moa.user.UserListSnapshots
```

public class UserListSnapshots extends Request

A request to list the snapshots

15.20.6 Constructor Index

UserListSnapshots()

UserListSnapshots(String, int, String)

15.20.7 Method Index

getHost() The host computer

getLaunchName() The launch name of the agent

getPort() The port on host computer

15.20.8 Constructors

UserListSnapshots public UserListSnapshots()

UserListSnapshots public UserListSnapshots(String h, int p, String launchName)

Parameters:

h - the host computer

p - the port on the host

launchName - the name of the launched agent

15.20.9 Methods

public String getHost()

The host computer

public int getPort()

The port on host computer

public String getLaunchName()

The launch name of the agent

15.21 UserLogQuery

```
java.lang.Object
   |
   +----ORG.opengroup.moa.user.Request
           |
           +----ORG.opengroup.moa.user.UserLogQuery
```

public class UserLogQuery extends Request

A request for a log query

15.21.1 Constructor Index

UserLogQuery()

UserLogQuery(String, int, Object)

15.21.2 Method Index

getHost() The host computer

getLogQuery() The query object

getPort() The port on the host

15.21.3 Constructors

UserLogQuery public UserLogQuery()

UserLogQuery public UserLogQuery(String h, int p, Object f)

Parameters:

- h - the host computer
- p - the port on the host
- f - the query

15.21.4 Methods

public Object getLogQuery()

The query object

public String getHost()

The host computer

public int getPort()

The port on the host

16 sandbox

16.1 Cache

```
java.lang.Object
   |
   +----java.lang.Thread
           |
           +----ORG.opengroup.sandbox.Cache
```

public class Cache extends Thread

16.1.1 Constructor Index

Cache(long, int)

16.1.2 Method Index

add(Cachable)

getEntries() Return the current number of entries of the cache.

getTime()

remove(Cachable)

reschedule(Cachable)

run()

16.1.3 Constructors

Cache public Cache(long granularity, int extent)

16.1.4 Methods

public long getTime()

public void run()

Overrides:

run in class Thread

public synchronized void add(Cachable c)

public synchronized void reschedule(Cachable c)

public void remove(Cachable c)

public int getEntries()

Return the current number of entries of the cache. Even though the entries is maintained in the Cache object, it is up to the object that adds/removes entries to act upon this information.

16.2 CallBack

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.CallBack
```

public final class CallBack extends Object implements SandboxListener

Provides a context for requests and events.

16.2.1 Constructor Index

CallBack(CallBackListener) Create a new CallBack.

16.2.2 Method Index

bump(int, int) Bump the count and check the limit on the sandbox.

processEvent(SandboxEvent) Submits a sandbox event.

processRequest(Object, long) Fires a SandBoxRequest event and waits for a response.

processRequest(SandboxRequest) Submits an asynchronous sandbox request event.

16.2.3 Methods

public CallBack(CallBackListener listener)

Create a new CallBack.

public void processEvent(SandboxEvent se)

Submits a sandbox event.

public void processRequest(SandboxRequest sr)

Submits an asynchronous sandbox request event.

public Object processRequest(Object eventItem, long timeout)

Fires a SandBoxRequest event and waits for a response. A failure is defined as a null return, in which case the reason of the failure has been appended to the whyNot parameter.

Parameters:

eventItem - defines the client's request.

timeout - maximum response time, in milliseconds.

Returns:

the response to the request, or a Failure object.

public void bump(int ndx, int inc)

Bump the count and check the limit on the sandbox.

16.3 Failure

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.Failure
```

public final class Failure extends Object implements Serializable

Signals that a failure has occurred and gives the reason for the failure. Failure objects may be returned as a result of a sandbox request.

16.3.1 Field Index

reason This is the reason for the failure.

16.3.2 Constructor Index

Failure(String)

16.3.3 Method Index

toString() Converts Failure to a String.

16.3.4 Fields

reason public final String reason

This is the reason for the failure.

16.3.5 Constructors

Failure public Failure(String why)

Parameters:

why - the reason for the failure.

16.3.6 Methods

public String toString()

Converts Failure to a String.

Overrides:

toString in class Object

16.4 Queue

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.Queue
```

public class Queue extends Object implements Serializable

16.4.1 Constructor Index

Queue()

16.4.2 Method Index

dequeue() Fetch from the queue.

empty() Empty the queue.

enqueue(Object) Add to the queue.

getCapacity() Returns the capacity of the queue at the given time.

getSize() Returns the size of the queue at the given time.

isEmpty() Check for empty queue.

16.4.3 Constructors

Queue public Queue()

16.4.4 Methods

public synchronized void enqueue(Object obj)

Add to the queue.

public synchronized Object dequeue() throws InterruptedException

Fetch from the queue.

public synchronized boolean isEmpty()

Check for empty queue.

public synchronized void empty()

Empty the queue.

public synchronized int getCapacity()

Returns the capacity of the queue at the given time.

public synchronized int getSize()

Returns the size of the queue at the given time.

16.5 RequestControl

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.RequestControl
```

public class RequestControl extends Object

Provides the context for a request.

16.5.1 Field Index

consumed

expire

The time when the request will expire.

sb

thePartialResult

theResult

16.5.2 Constructor Index

RequestControl(long)

16.5.3 Method Index

bump(int, int) Bump the count and check the limit on the sandbox.

isConsumed() Returns true if there has been a response (success or failure) or if the request has timed out.

partialResult(Object) Used to return the partial results of the request.

result(Object) Used to return the results of the request.

16.5.4 Fields

expire public final long expire

The time when the request will expire.

consumed protected boolean consumed

theResult protected Object theResult

thePartialResult protected Object thePartialResult

sb protected Sandbox sb

16.5.5 Constructors

public RequestControl(long time)

Parameters:

time - when the request will expire (in milliseconds).

16.5.6 Methods

public boolean isConsumed()

Returns true if there has been a response (success or failure) or if the request has timed out. Overridden by CallBack.

public void result(Object v)

Used to return the results of the request. A failure is indicated by passing a Failure object.

public final void partialResult(Object v)

Used to return the partial results of the request. The difference between result and partialResult is that any number of partial results may be given (only the last one is used), but the first result call consumes the event and no further results or partial results will be accepted. Partial results are used in place of a final result when the result is a timeout and the partial result is not null.

public void bump(int ndx, int inc)

Bump the count and check the limit on the sandbox.

16.6 Sandbox

This package provides the basis for the trust domains that are needed to allow trusted code to interact with untrusted code. This package is not specific to the MOA system though it is used extensively by it. It can be used by another system that needs its features.

The sandbox (and callback) manage the passing of events between threads.

Sandbox also manages the activity counts which help in debugging.

One of the key considerations governing all moa component interaction is trust.

Each agent has its own trust domain. The agent environment may also be one or more trust domains. Each trust domain has its own set of threads, organized as a thread group or sandbox.

Events and requests (two-way events), when passed across trust domains, must be handed off from the thread of one trust domain to the thread of the other trust domain. This is handled by CallBack objects. There are two advantages to doing this:

1.When trusted code passes an event or request to untrusted code, the untrusted code is not given access to the thread belonging to the trusted code. The consequence of this is that it is possible to terminate all the threads belonging to the untrusted code and be sure that the untrusted code is not executing on a"borrowed" thread.

2.When a SecurityManager is used to restrict what untrusted code is allowed to do, it works by examining the stack of a thread. So long as there is any untrusted code on the thread stack, no operations requiring trust are allowed to execute. However, untrusted code can, using thread switching, pass a request to trusted code.

A CallBack object defines a client/server relationship (on the same jvm). A server authors a CallBack object and passes the CallBack to a client. (Or the server may publish the CallBack so that potential clients have access to it.) Once a client receives a CallBack, the client can then use the CallBack to send events or make a request.

Trust domains in MOA also have associated activity counts, each service having its own counters which it can increment on the trust domain requesting the service. The intent of these counts is to allow for the monitoring of agent behavior.

16.7 Sandbox

```
java.lang.Object
   |
   +----java.lang.ThreadGroup
           |
           +----ORG.opengroup.sandbox.Sandbox
```

public class Sandbox extends ThreadGroup

A sandbox is used to move client events and requests onto server threads. This is done either because the server is not to be trusted with a client thread, or because the client thread is not trusted enough to handle a request. A second use of a sandbox is to provide a synchronous api on an asynchronous service.

16.7.1 Field Index

COUNT_SIZE Size of counts and limits arrays needed by Sandbox.

DST_ASYNC_REQUESTS Index for count of requests that went across sandboxes, were submitted asynchronously (no wait for response), and this sandbox was the destination.

DST_EVENTS Index for count of events that went across sandboxes and this sandbox was the destination.

DST_SYNC_REQUESTS Index for count of requests that went across sandboxes, were submitted synchronously (wait for response), and this sandbox was the destination.

SHORT_ASYNC_REQUESTS Index for count of requests for the same sandbox and were submitted asynchronously (no wait for response).

SHORT_EVENTS Index for count of events for the same sandbox.

SLF_ASYNC_REQUESTS Index for count of requests for the same sandbox and were submitted asynchronously (no wait for response).

SLF_EVENTS Index for count of events for the same sandbox.

SLF_SYNC_REQUESTS Index for count of requests submitted synchronously (wait for response) for the same sandbox.

SRC_ASYNC_REQUESTS Index for count of requests that went across sandboxes, were submitted asynchronously (no wait for response), and this sandbox was the source.

SRC_EVENTS Index for count of events that went across sandboxes and this sandbox was the source.

SRC_SYNC_REQUESTS Index for count of requests that went across sandboxes, were submitted synchronously (wait for response), and this sandbox was the source.

TIMEOUT_ASYNC_REQUESTS Index for count of requests submitted asynchronously (no wait for response) which timed out.

TIMEOUT_SYNC_REQUESTS Index for count of requests submitted synchronously (wait for response) which timed out.

16.7.2 Constructor Index

Sandbox(String, long, int, int)

16.7.3 Method Index

addLimitEventListener(CallBack) Register for limit events.

bump(int, int) Bump the count and check the limit.

clear(Failure) Clears all callbacks and aborts any pending requests.

currentSandbox() Returns the Sandbox for the current thread.

getCapacity() Returns the capacity of the queue at the given time.

getCount(int) Returns the selected count.

getLimit(int) Returns the selected limit.

getTotalCountSize() removeLimitEventListener(CallBack)

Unregister for limit events.

setDescriptions(String[]) Initialize the elements of an array of count descriptions.

setLimit(int, int)

setTotalCountSize(int) Set the size of the counts and limits arrays to be maintained by Sandbox.

16.7.4 Fields

SRC_EVENTS public static final int SRC_EVENTS

Index for count of events that went across sandboxes and this sandbox was the source.

DST_EVENTS public static final int DST_EVENTS

Index for count of events that went across sandboxes and this sandbox was the destination.

SLF_EVENTS public static final int SLF_EVENTS

Index for count of events for the same sandbox. (The sandbox was not short-circuited--the event was not queued.)

SHORT_EVENTS public static final int SHORT_EVENTS

Index for count of events for the same sandbox. (The sandbox was short-circuited, as the Cache thread was being used.)

SRC_ASYNC_REQUESTS public static final int SRC_ASYNC_REQUESTS

Index for count of requests that went across sandboxes, were submitted asynchronously (no wait for response), and this sandbox was the source.

DST_ASYNC_REQUESTS public static final int DST_ASYNC_REQUESTS

Index for count of requests that went across sandboxes, were submitted asynchronously (no wait for response), and this sandbox was the destination.

SLF_ASYNC_REQUESTS public static final int SLF_ASYNC_REQUESTS

Index for count of requests for the same sandbox and were submitted asynchronously (no wait for response). (The sandbox was not short-circuited, as the Cache thread was being used.)

SHORT_ASYNC_REQUESTS public static final int SHORT_ASYNC_REQUESTS

Index for count of requests for the same sandbox and were submitted asynchronously (no wait for response). (The sandbox was short-circuited--the request was not queued.)

TIMEOUT_ASYNC_REQUESTS public static final int TIMEOUT_ASYNC_REQUESTS

Index for count of requests submitted asynchronously (no wait for response) which timed out.

SRC_SYNC_REQUESTS public static final int SRC_SYNC_REQUESTS

Index for count of requests that went across sandboxes, were submitted synchronously (wait for response), and this sandbox was the source.

DST_SYNC_REQUESTS public static final int DST_SYNC_REQUESTS

Index for count of requests that went across sandboxes, were submitted synchronously (wait for response), and this sandbox was the destination.

SLF_SYNC_REQUESTS public static final int SLF_SYNC_REQUESTS

Index for count of requests submitted synchronously (wait for response) for the same sandbox. (However, he sandbox was not short-circuted, as the originator waits for a response.)

TIMEOUT_SYNC_REQUESTS public static final int TIMEOUT_SYNC_REQUESTS

Index for count of requests submitted synchronously (wait for response) which timed out.

COUNT_SIZE public static final int COUNT_SIZE

Size of counts and limits arrays needed by Sandbox.

16.7.5 Constructors

Sandbox public Sandbox(String name, long granularity, int extent, int serverThreads)

Parameters:

name - the name of the thread group.

granularity - length of sleep of Cache.

extend - size of Cache array.

serverThreads - number of threads created.

aggregateSize - size of totals and limits arrays.

16.7.6 Methods

public static final void setTotalCountSize(int count)

Set the size of the counts and limits arrays to be maintained by Sandbox. This method must be called only once, and prior to creating the first instance of sandbox.

public static final int getTotalCountSize()

public static final void setDescriptions(String d[])

Initialize the elements of an array of count descriptions.

public static Sandbox currentSandbox()

Returns the Sandbox for the current thread.

public void clear(Failure reason)

Clears all callbacks and aborts any pending requests.

Parameters:

reason - is returned to all pending and subsequent requests.

public int getCapacity()

Returns the capacity of the queue at the given time.

public long getCount(int ndx)

Returns the selected count.

public long getLimit(int ndx)

Returns the selected limit.

public void setLimit(int ndx, int v)

public void addLimitEventListener(CallBack cb) throws TooManyListenersException

Register for limit events.

public void removeLimitEventListener(CallBack cb)

Unregister for limit events.

public static void bump(int ndx, int inc)

Bump the count and check the limit.

16.8 SandboxEvent

```
java.lang.Object
  |
  +----java.util.EventObject
          |
          +----ORG.opengroup.sandbox.SandboxEvent
```

public final class SandboxEvent extends EventObject

Generic Sandbox Event.

16.8.1 Field Index

eventItem Event-specific item.

16.8.2 Constructor Index

SandboxEvent(Object, Object) Provides context for callback.

16.8.3 Fields

eventItem public final Object eventItem

Event-specific item.

16.8.4 Constructors

SandboxEvent public SandboxEvent(Object source, Object item)

Provides context for callback. May be null.

16.9 SandboxRequest

```
java.lang.Object
  |
  +----java.util.EventObject
          |
          +----ORG.opengroup.sandbox.SandboxRequest
```

public final class SandboxRequest extends EventObject

16.9.1 Field Index

eventItem Event-specific item.

requestControl Provides context for response.

16.9.2 Constructor Index

SandboxRequest(Object, Object, RequestControl)

16.9.3 Method Index

bump(int, int) Bump the count and check the limit on the sandbox.

isConsumed() Returns true if there has been a response (success or failure) or if the request has timed out.

partialResult(Object) Used to return the partial results of the request.

result(Object) Used to return the results of the request.

16.9.4 Fields

eventItem public final Object eventItem

Event-specific item.

requestControl public final RequestControl requestControl

Provides context for response.

16.9.5 Constructors

SandboxRequest public SandboxRequest(Object source, Object item, RequestControl rc)

Parameters:

source - originator of the request.

item - definition of the request

rc - facility for respond.

16.9.6 Methods

public boolean isConsumed()

Returns true if there has been a response (success or failure) or if the request has timed out.

public void result(Object v)

Used to return the results of the request. A failure is indicated by passing a Failure object.

public final void partialResult(Object v)

Used to return the partial results of the request. The difference between result and partialResult is that any number of partial results may be given (only the last one is used), but the first result call consumes the event and no further results or partial results will be accepted. Partial results are used in place of a final result when the result is a timeout and the partial result is not null.

public void bump(int ndx, int inc)

Bump the count and check the limit on the sandbox.

16.10 SandboxSupport

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.SandboxSuppor
```

public class SandboxSupport extends Object implements Serializable

Handles listener registration and event firing.

16.10.1 Constructor Index

SandboxSupport(Object)

16.10.2 Method Index

addSandboxListener(SandboxListener) Register for sandbox events and requests.

fireEvent(Object) Fires a Sandbox Event

fireEvent(Object, RequestControl) Fires a Sandbox Request Event

fireEvent(SandboxEvent) Fires a Sandbox Event

fireEvent(SandboxRequest) Fires a Sandbox Request Event

hasListeners() Returns true when there are registered listeners.

removeAll() Removes all listeners.

removeSandboxListener(SandboxListener) Unregister for sandbox events and requests.

16.10.3 Constructors

SandboxSupport public SandboxSupport(Object sourceBean)

16.10.4 Methods

public void fireEvent(Object item)

Fires a Sandbox Event

public void fireEvent(SandboxEvent e)

Fires a Sandbox Event

public void fireEvent(Object item, RequestControl requestControl)

Fires a Sandbox Request Event

public void fireEvent(SandboxRequest e)

Fires a Sandbox Request Event

public void addSandboxListener(SandboxListener listener)

Register for sandbox events and requests.

public void removeSandboxListener(SandboxListener listener)

Unregister for sandbox events and requests.

public void removeAll()

Removes all listeners.

public boolean hasListeners()

Returns true when there are registered listeners.

16.11 Cachable (Interface)

public interface Cachable

A cachable item is retained as long as it is used occasionally.

See Also:

Cache

16.11.1 Method Index

getEntry() Get the entry for the cachable, needed for remove; this field is meaningful only inside the Cache object.

lastUsed() Returns time of last use in milliseconds GMT.

release() Release the item, it is not in active use.

retain() Returns the number of milliseconds the item is to be retained after its last use.

setEntry(int) Set the entry for the cachable, needed for remove; this field is meaningful only inside the Cache object.

16.11.2 Methods

public abstract long lastUsed()

Returns time of last use in milliseconds GMT.

See Also:

currentTimeMillis

public abstract long retain()

Returns the number of milliseconds the item is to be retained after its last use.

public abstract void release()

Release the item, it is not in active use.

public abstract void setEntry(int entry)

Set the entry for the cachable, needed for remove; this field is meaningful only inside the Cache object.

public abstract int getEntry()

Get the entry for the cachable, needed for remove; this field is meaningful only inside the Cache object.

16.12 CallBackListener (Interface)

public interface CallBackListener extends SandboxListener

Receive events, requests, and request timeouts.

16.12.1 Method Index

requestTimeout(RequestControl) Notifies the server that a request has timed out.

16.12.2 Methods

public abstract void requestTimeout(RequestControl rc)

Notifies the server that a request has timed out.

16.13 SandboxListener (Interface)

public interface SandboxListener extends EventListener

Listener on the sandbox for sandbox requests and events.

16.13.1 Method Index

processEvent(SandboxEvent) Submits a sandbox event.

processRequest(SandboxRequest) Submits an asynchronous sandbox request event.

16.13.2 Methods

public abstract void processEvent(SandboxEvent se)

Submits a sandbox event.

public abstract void processRequest(SandboxRequest sr)

Submits an asynchronous sandbox request event.

17 sandbox.environment

This package builds on the ORG.opengroup.sandbox package to provide a thin container for objects that are instances of classes that extend EnvironmentChild. Thus it provides the object tree for a system that chooses to use the sandbox and allows it to be identified by a distinctly unique path. Killer is a utility class that allows the system using the sandbox to be terminated. Main is a simple bean container for holding environment objects.

17.1 Environment

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.environment.Environment
```

public class Environment extends EnvironmentChild

17.1.1 Field Index

ADDED Identifier for EnvironmentEvent.getTypeName().

REMOVED Identifier for EnvironmentEvent.getTypeName().

17.1.2 Constructor Index

Environment()

17.1.3 Method Index

addChild(String, Object) Add a child.

addChildren(EnvironmentChild[]) Add children and wire.

addEnvironmentListener(EnvironmentListener)

getComponentCount()

getComponentName()

getComponentName(int)

getInstanceOf(Class) Returns an object representing a specified type view.

getInstanceOf(String, Class) Returns an object reperesenting a specified type view of the named component.

init() This code is called by EnvironmentChild

isInstanceOf(Class) Returns true if this component can be viewed as a given target type.

isInstanceOf(String, Class) Returns true if the named component can be viewed as a given target type.

removeChildren(String[]) Remove children.

removeEnvironmentListener(EnvironmentListener)

setComponentCount(int)

setComponentName(int, String)

setComponentName(String[])

17.1.4 Fields

ADDED public static final String ADDED

Identifier for EnvironmentEvent.getTypeName().

REMOVED public static final String REMOVED

Identifier for EnvironmentEvent.getTypeName().

17.1.5 Constructors

Environment public Environment()

17.1.6 Methods

getComponentCount public final int getComponentCount()

setComponentCount public final void setComponentCount(int s)

getComponentName public final String[] getComponentName()

setComponentName public final void setComponentName(String a[])

getComponentName public final String getComponentName(int i) throws ArrayIndexOutOfBoundsException

setComponentName public final void setComponentName(int i, String s) throws ArrayIndexOutOfBoundsException

init public void init()

This code is called by EnvironmentChild

Overrides:

init in class EnvironmentChild

addEnvironmentListener public final void addEnvironmentListener(EnvironmentListener el)

removeEnvironmentListener public final void removeEnvironmentListener(EnvironmentListener el)

addChild public final void addChild(String name, Object obj)

Add a child. Set the name and wire it if it is an EnvironmentChild.

addChildren public final void addChildren(EnvironmentChild ec[])

Add children and wire. (setName() has previously been called.)

removeChildren public final void removeChildren(String name[])

isInstanceOf public final synchronized boolean isInstanceOf(Class targetType)

Returns true if this component can be viewed as a given target type.

Parameters:

targetType - The type of view we would like to see

Overrides:

isInstanceOf in class EnvironmentChild

isInstanceOf public final synchronized boolean isInstanceOf(String name, Class targetType)

Returns true if the named component can be viewed as a given target type.

Parameters:

name - Identifies the bean held by the environment hash table.

targetType - The type of view we'd like to get.

getInstanceOf public final synchronized Object getInstanceOf(Class targetType)

Returns an object representing a specified type view. Expected usage is of the form:

(TargetType)getInstanceOf(TargetType.class) which will raise a casting exception if the derived class is not an instance of TargetType.

Parameters:

targetType - The type of view we would like to see

Overrides:

getInstanceOf in class EnvironmentChild

getInstanceOf public final synchronized Object getInstanceOf(String name, Class targetType)

Returns an object reperesenting a specified type view of the named component.

Parameters:

name - Identifies the bean held by the environment hash table.

targetType - The type of view we'd like to get.

17.2 EnvironmentChild

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
```

public class EnvironmentChild extends Object implements Serializable

17.2.1 Field Index

DELIM Component name delimiter.

17.2.2 Constructor Index

EnvironmentChild()

17.2.3 Method Index

getEnvironment() Returns the context of the bean.

getFullName() Returns the full name of the bean

getInstanceOf(Class) Returns an object representing a specified type view.

getName() Returns the name of the bean

getPath() Returns the path of the bean

getPrivateSandboxCacheDelay() getPrivateSandboxCacheExtent()

getSandbox()

getSandboxThreads()

init() The user extending this class is expected to overwrite this class and provide desiredfunctionality for initializations

isInstanceOf(Class) Returns true if this component can be viewed as a given target type.

isMakePrivateSandbox()

isWired() Returns true if the bean has been wired.

setContext(Environment) This method takes care of the requirements of the first phase by registering the component and associating it with an environment.

setMakePrivateSandbox(boolean) This property can be set only at design time.

setName(String) Set the name of the bean

setPrivateSandboxCacheDelay(int) This property can only be set at design time.

setPrivateSandboxCacheExtent(int) This property can only be set at design time.

setSandboxThreads(int) This property can only be set at design time.

wire() This method is called after setContext() Its purpose is to allow the component to wire itself to the other components that it needs in the environment The functions returns if wired is set to true, otherwise it sets it to true.

17.2.4 Fields

DELIM public static String DELIM

Component name delimiter.

17.2.5 Constructors

EnvironmentChild public EnvironmentChild()

17.2.6 Methods

public final boolean isWired()

Returns true if the bean has been wired.

public final String getName()

Returns the name of the bean

public final String getFullName()

Returns the full name of the bean

public final String getPath()

Returns the path of the bean

public final void setName(String v)

Set the name of the bean

public final Environment getEnvironment()

Returns the context of the bean.

public final boolean isMakePrivateSandbox()

public final void setMakePrivateSandbox(boolean makePrivate)

This property can be set only at design time.

Parameters:

makePrivate. - If the EnvironmentChild object wants its own sandbox, this should be set to true, otherwise false Default value is false

public final int getSandboxThreads()

public final void setSandboxThreads(int v)

This property can only be set at design time.

public final int getPrivateSandboxCacheDelay()

public final void setPrivateSandboxCacheDelay(int delay)

This property can only be set at design time.

public final int getPrivateSandboxCacheExtent()

public final void setPrivateSandboxCacheExtent(int extent)

This property can only be set at design time.

public final void setContext(Environment env)

This method takes care of the requirements of the first phase by registering the component and associating it with an environment. It has to be called before wire()

Parameters:

env - Gives the component a reference to the environment. The component called is then expected to wire itself (via the wire() method) together with the various components on which it depends

public Sandbox getSandbox()

public final void wire()

This method is called after setContext() Its purpose is to allow the component to wire itself to the other components that it needs in the environment The functions returns if wired is set to true, otherwise it sets it to true.

public void init()

The user extending this class is expected to overwrite this class and provide desired functionality for initializations

public boolean isInstanceOf(Class targetType)

Returns true if this component can be viewed as a given target type. This method needs to be overridden only if the derived class supports aggregation.

Parameters:

targetType - The type of view we would like to see

public Object getInstanceOf(Class targetType)

Returns an object representing a specified type view. This method needs to be overridden only if the derived class supports aggregation. Expected usage is of the form:

(TargetType)getInstanceOf(TargetType.class) which will raise a casting exception if the derived class is not an instance of TargetType

Parameters:

targetType - The type of view we would like to see

17.3 EnvironmentEvent

```
java.lang.Object
  |
  +----java.util.EventObject
          |
          +----ORG.opengroup.sandbox.environment.EnvironmentEvent
```

public final class EnvironmentEvent extends EventObject

Describes changes to nesting state.

17.3.1 Method Index

childrenAdded(Environment, Object[])

childrenRemoved(Environment, Object[])

getChildren()

getTypeName()

isChild(Object)

17.3.2 Methods

public static EnvironmentEvent childrenAdded(Environment source, Object kids[])

public static EnvironmentEvent childrenRemoved(Environment source, Object kids[])

public String getTypeName()

public Object[] getChildren()

public boolean isChild(Object child)

17.4 Killer

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.environment.Killer
```

public class Killer extends EnvironmentChild implements Runnable

This class provides an extension of thread that fulfils our purpose of providing a definitive take down mechanism for the Juggler

17.4.1 Constructor Index

Killer()

17.4.2 Method Index

getTimeout()

init() The user extending this class is expected to overwrite this class and provide desired functionality for initializations

printSandboxStats()

run()

setTimeout(long)

takeDown()

17.4.3 Constructors

Killer public Killer()

17.4.4 Methods

public void setTimeout(long t)

public long getTimeout()

public void init()

The user extending this class is expected to overwrite this class and provide desired functionality for initializations

Overrides:

init in class EnvironmentChild

public void run()

public void takeDown()

public void printSandboxStats()

17.5 Main

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
          |
          +----ORG.opengroup.sandbox.environment.Killer
```

public class Killer extends EnvironmentChild implements Runnable

This class provides an extension of thread that fulfils our purpose of providing a definitive take down mechanism for the Juggler

17.5.1 Constructor Index

Killer()

17.5.2 Method Index

getTimeout()

init() The user extending this class is expected to overwrite this class and provide desired functionality for initializations

printSandboxStats()

run()

setTimeout(long)

takeDown()

17.5.3 Constructors

Killer public Killer()

17.5.4 Methods

public void setTimeout(long t)

public long getTimeout()

public void init()

The user extending this class is expected to overwrite this class and provide desired functionality for initializations

Overrides:

init in class EnvironmentChild

public void run()

public void takeDown()

public void printSandboxStats()

17.6 Service

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.Main
```

public class Main extends Object

The Main class is a very simple bean container which instantiates a bean using java.beans.Beans.instantiate. Main takes one command line parameter, the name of the bean to be instantiated.

17.6.1 Constructor Index

Main()

17.6.2 Method Index

exit()

main(String[]) Program starting point.

17.6.3 Constructors

Main public Main()

17.6.4 Methods

public static void main(String args[])

Program starting point.

public static void exit()

17.7 EnvironmentListener (Interface)

public interface EnvironmentListener extends EventListener

Receives EnvironmentEvent's.

17.7.1 Method Index

nestedStateChanged(EnvironmentEvent) Nesting state has changed.

17.7.2 Methods

public abstract void nestedStateChanged(EnvironmentEvent se)

Nesting state has changed.

18 sandbox.message

This package builds on the ORG.opengroup.sandbox.net packages channels to provide a messaging interface to systems using the sandbox model. The messenger delivers messages across the network. Messages have a fixed max length. Deserialization of content is delayed until requested, allowing the requestor to specify the class loader to be used. Messages are passed using SandboxEvent.

18.1 ChannelManager

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.net.NetUser
                   |
                   +----ORG.opengroup.sandbox.message.ChannelManager
```

public class ChannelManager extends NetUser

A bean which manages all channels

18.1.1 Constructor Index

ChannelManager()

18.1.2 Method Index

getOpenChannelTimeout()

getSendMessageTimeout() Return the send message timeout

init() Initialize.

processRequest(SandboxRequest) Overrides NetUser.processRequest for incoming channels.

setOpenChannelTimeout(int)

setSendMessageTimeout(int) Set send message timeout

18.1.3 Constructors

ChannelManager public ChannelManager()

18.1.4 Methods

public void init()

Initialize.

Overrides:

init in class NetUser

public final void setSendMessageTimeout(int v)

Set send message timeout

public final int getSendMessageTimeout()

Return the send message timeout

public void setOpenChannelTimeout(int tmo)

public int getOpenChannelTimeout()

public void processRequest(SandboxRequest channelRequest)

Overrides NetUser.processRequest for incoming channels.

Overrides:

processRequest in class NetUser

18.2 Message

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.net.ChannelDescriptor
           |
           +----ORG.opengroup.sandbox.message.Message
```

public final class Message extends ChannelDescriptor

Passed as an event or a request between systems.

18.2.1 Field Index

content

hops The content being passed.

MAX_SIZE

seq

18.2.2 Constructor Index

Message(String, String, int, String, String, int, byte, int, int, byte[])

Message(String, String, int, String, String, int, byte, int, int, Object)

18.2.3 Method Index

getContent(ClassLoader)

18.2.4 Fields

MAX_SIZE public final int MAX_SIZE

hops public final int hops

The content being passed.

seq public final int seq

content public final byte content[]

18.2.5 Constructors

Message public Message(String origNam, String origHost, int origPort, String destNam, String destHost, int destPort, byte att, int hops, int seq, byte pcontent[]) throws SizeException

Message public Message(String origNam, String origHost, int origPort, String destNam, String destHost, int destPort, byte att, int hops, int seq, Object pcontent) throws IOException, SizeException

18.2.6 Methods

public Object getContent(ClassLoader cl) throws IOException, ClassNotFoundException

18.3 MessageUser

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
        |
        +----ORG.opengroup.sandbox.message.MessageUser
```

public class MessageUser extends EnvironmentChild

18.3.1 Field Index

messenger A reference to the Messenger instance

18.3.2 Constructor Index

MessageUser()

18.3.3 Method Index

getClassLoader(Message)

getHost() Returns the local host

getIncomingMessageListener() Implementation of the MessengerUser interface method

getMessengerBean() Return the name of the messenger bean

getPort() Returns the local port

incomingMessage(Message) Called for incoming messages.

incomingRequest(String, String, int, byte, Object) Called for incoming request messages.

incomingXRequest(String, String, int, byte, Object, String) Called for incoming request messages.

init() Bean initialization

isReceiverPresent(byte) AC overrides this method and checks if the att indicates that the request is for agent and if agent is present.

isSuspended()

resume()

sendMessage(Message) send a message

sendMessage(String, String, int, byte, int, Object) Sends a message, returning null if successful.

sendRequest(String, String, int, byte, Object, int) Sends a request message, returning Failure if unsuccessful.

sendRequests(String, String, int, byte, int, Object, int, AEname[]) Sends a request message, returning Failure if unsuccessful.

setMessengerBean(String) Set the name of the messenger bean

suspend()

18.3.4 Fields

messenger protected transient Messenger messenger

A reference to the Messenger instance

18.3.5 Constructors

MessageUser public MessageUser()

18.3.6 Methods

public final void setMessengerBean(String messengerBean)

Set the name of the messenger bean

public final String getMessengerBean()

Return the name of the messenger bean

public final CallBack getIncomingMessageListener()

Implementation of the MessengerUser interface method

public void init()

Bean initialization

Overrides:

init in class EnvironmentChild

protected void incomingMessage(Message m)

Called for incoming messages. Must be overridden.

public Object incomingXRequest(String origName, String origHost, int origPort, byte att, Object content, String destName)

Called for incoming request messages. Must be overridden.

public Object incomingRequest(String origName, String origHost, int origPort, byte att,

Object content)

Called for incoming request messages. Must be overridden.

public boolean isReceiverPresent(byte att)

AC overrides this method and checks if the att indicates that the request is for agent and if agent is present. If so, it returns true. It returns false only in case that agent is absent and request was indeed for the agent.

protected Failure sendMessage(String destNam, String destHost, int destPort, int hops,

Object content)

Sends a message, returning null if successful.

Parameters:

name - identifies the requesting agent

destName - receiving agent

destHost - destination host

destPort - destination port number

att - Opaque attribute

content - Object being passed

public Object sendRequest(String destNam, String destHost, int destPort, byte att,

Object content, int tmo)

Sends a request message, returning Failure if unsuccessful. this form is used for stationary agents!

Parameters:

name - identifies the requesting agent

destName - receiving agent

destHost - destination host

destPort - destination port number

content - Object being passed

att - Opaque attribute

tmo - timeout delay

public Object sendRequests(String destNam, String destHost, int destPort, byte att, int hops, Object content, int tmo, AEname itinerary[])

Sends a request message, returning Failure if unsuccessful.

Parameters:

name - identifies the requesting agent

destName - receiving agent

destHost - destination host

destPort - destination port number

hops - number of hops allowed

hops - number of hops allowed

content - Object being passed

att - Opaque attribute

tmo - timeout delay

array - of URLS where to search for agent

protected Failure sendMessage(Message m)

send a message

public String getHost()

Returns the local host

public int getPort()

Returns the local port

public synchronized boolean suspend()

public synchronized boolean resume()

public synchronized boolean isSuspended()

protected ClassLoader getClassLoader(Message m)

18.4 Messenger

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
          |
          +----ORG.opengroup.sandbox.net.NetUser
                  |
                  +----ORG.opengroup.sandbox.message.ChannelManager
```

```
|
+----ORG.opengroup.sandbox.message.Messenger
```

public class Messenger extends ChannelManager

Handles message passing.

18.4.1 Field Index

COUNT_SIZE Size of counts and limits arrays needed by Messenger.

MESSAGE_BYTES_READ Index for count of message bytes read.

MESSAGE_BYTES_WRITTEN Index for count of message bytes written.

MESSAGES_READ Index for count of messages read.

MESSAGES_WRITTEN Index for count of messages written.

18.4.2 Constructor Index

Messenger()

18.4.3 Method Index

addMessageListener(CallBack) Register for message notifications.

removeMessageListener(CallBack) Unregister for message notifications.

setDeadMessageReceiver(DeadMessageReceiver) Specify the processor of rejected or undeliverable messages.

setDescriptions(String[]) Initialize the elements of an array of count descriptions.

18.4.4 Fields

MESSAGES_READ public static final int MESSAGES_READ

Index for count of messages read.

MESSAGE_BYTES_READ public static final int MESSAGE_BYTES_READ

Index for count of message bytes read.

MESSAGES_WRITTEN public static final int MESSAGES_WRITTEN

Index for count of messages written.

MESSAGE_BYTES_WRITTEN public static final int MESSAGE_BYTES_WRITTEN

Index for count of message bytes written.

COUNT_SIZE public static final int COUNT_SIZE

Size of counts and limits arrays needed by Messenger.

18.4.5 Constructors

Messenger

public Messenger()

18.4.6 Methods

public static final void setDescriptions(String d[])

Initialize the elements of an array of count descriptions.

public void addMessageListener(CallBack listener) throws TooManyListenersException.

Register for message notifications.

public void removeMessageListener(CallBack listener)

Unregister for message notifications.

public void setDeadMessageReceiver(DeadMessageReceiver receiver)

Specify the processor of rejected or undeliverable messages.

18.5 ObjInStr

```
java.lang.Object
   |
   +----java.io.InputStream
           |
           +----java.io.ObjectInputStream
                   |
                   +----ORG.opengroup.sandbox.message.ObjInStr
```

public class ObjInStr extends ObjectInputStream

Extends ObjectInputStream by allowing explicit specification of the class loader.

18.5.1 Constructor Index

ObjInStr(InputStream, ClassLoader)

18.5.2 Method Index

resolveClass(ObjectStreamClass)

18.5.3 Constructors

ObjInStr public ObjInStr(InputStream is, ClassLoader pcl) throws IOException

18.5.4 Methods

protected Class resolveClass(ObjectStreamClass v) throws IOException, ClassNotFoundException

Overrides:

resolveClass in class ObjectInputStream

18.6 ReliableContent

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.message.ReliableContent
```

public class ReliableContent extends Object implements Serializable

18.6.1 Constructor Index

ReliableContent()

18.6.2 Constructors

public ReliableContent()

18.7 DeadMessageReceiver (Interface)

public interface DeadMessageReceiver

Handles rejected and undeliverable messages.

18.7.1 Method Index

noReceiver(Message) Receiver does not exist on this host, try to forward the message

suspendedReceiver(Message) Receiver is suspended, if AC, deliver message to monitor

18.7.2 Methods

public abstract void noReceiver(Message message)

Receiver does not exist on this host, try to forward the message

public abstract Failure suspendedReceiver(Message message)

Receiver is suspended, if AC, deliver message to monitor

19 sandbox.net

This package provides a foundation for channel based communications between components that extend NetUser. Net is responsible for establishing channels between agents and various components, including messenger. It includes automatic retry capability for when an attempt at making a connection fails. (Not for when a connection that has been made fails!) It will also be responsible for securing the connections. (Key exchange using public keys.)

19.1 ChannelDescriptor

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.net.ChannelDescriptor
```

public class ChannelDescriptor extends Object implements Serializable

Describes a channel. When passed as an item in a SandboxRequest to net, the result is either a Failure or a ChannelAccess.

19.1.1 FieldIndex

attribute

destinationHost

destinationName

destinationPort

originatorHost

originatorName

originatorPort

19.1.2 Constructor Index

ChannelDescriptor(String, String, int, String, String, int, byte)

19.1.3 Fields

originatorName public final String originatorName

originatorHost public final String originatorHost

originatorPort public final int originatorPort

destinationName public final String destinationName

destinationHost public final String destinationHost

destinationPort public final int destinationPort

attribute public final byte attribute

19.1.4 Constructors

ChannelDescriptor

public ChannelDescriptor(String origNam, String origHost, int origPort, String destNam, String destHost, int destPort, byte att)

19.2 IncomingChannel

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.net.ChannelDescriptor
          |
          +----ORG.opengroup.sandbox.net.IncomingChannel
```

public final class IncomingChannel extends ChannelDescriptor

Passed as the item in a SandboxRequest from net when an incoming channel is received. Once a decision has been made to accept the channel or not, the response is sent via methods on response.

19.2.1 Constructor Index

IncomingChannel(String, String, int, String, String, int, byte, CallBackListener)

19.2.2 Method Index

accept(long) Returns the channel that was accepted.

accept(Object, RequestControl) Signals acceptance without waiting.

reject(Object, Failure) Rejects the incoming channel.

19.2.3 Constructors

IncomingChannel public IncomingChannel(String origNam, String origHost, int origPort, String destNam,String destHost, int destPort, byte att, CallBackListener sl)

19.2.4 Methods

public ChannelAccess accept(long timeout)

Returns the channel that was accepted.

public void accept(Object source, RequestControl rc)

Signals acceptance without waiting.

public void reject(Object source, Failure reason)

Rejects the incoming channel.

19.3 IncomingResponse

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.net.IncomingResponse
```

public final class IncomingResponse extends Object

Passed via an IncomingChannel.

19.3.1 Constructor Index

IncomingResponse(CallBackListener)

19.3.2 Method Index

accept(long) Returns the channel that was accepted.

accept(Object, RequestControl) Signals acceptance without waiting.

reject(Object, Failure) Rejects the incoming channel.

19.3.3 Constructors

IncomingResponse public IncomingResponse(CallBackListener sl)

19.3.4 Methods

accept public ChannelAccess accept(long timeout)

Returns the channel that was accepted.

accept public void accept(Object source, RequestControl rc)

Signals acceptance without waiting.

reject public void reject(Object source, Failure reason)

Rejects the incoming channel.

19.4 Net

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.net.Net
```

public class Net extends EnvironmentChild implements CallBackListener

A bean which manages all channels

19.4.1 Field Index

ABORTED_CHANNEL_REQUESTS Index for count of aborted channel requests.

BYTES_READ Index for count of bytes read.

BYTES_WRITTEN Index for count of bytes read.

CHANNEL_REQUESTS Index for count of channel requests.

COUNT_SIZE Size of counts and limits arrays needed by Net.

LOCAL_CHANNEL_ACCEPT Index for count of local channel attempts.

LOCAL_CHANNEL_ATTEMPT Index for count of local channel attempts.

LOCAL_CHANNEL_REQUEST_REJECT Index for count of local channel attempts.

LOCAL_CHANNEL_REQUEST_SUCCESS Index for count of local channel attempts.

REMOTE_CHANNEL_ACCEPT Index for count of local channel attempts.

REMOTE_CHANNEL_ATTEMPT Index for count of remote channel attempts.

REMOTE_CHANNEL_REQUEST_REJECT Index for count of local channel attempts.

REMOTE_CHANNEL_REQUEST_SUCCESS Index for count of local channel attempts.

19.4.2 Constructor Index

Net()

19.4.3 Method Index

addChannelListener(CallBack) Register for new channel notifications.

getCacheDelay()

getCacheExtent()

getExitOnServerError()

getHost()

getInitialRetryDelay()

getMaxRetryDelay()

getPort()

getRequestChannelCallBack()

Returns the CallBack used to pass channel requests.

getRetryDelayIncrement()

getRunServer()

getServerRetryDelay()

init() Extention of NetUser's init()

processEvent(SandboxEvent) Not used.

processRequest(SandboxRequest) Sandbox requests are passed to net to request a channel.

removeChannelListener(CallBack) Unregister for new channel notifications.

requestTimeout(RequestControl)

setCacheDelay(int) This property can only be set at design time.

setCacheExtent(int) This property can only be set at design time.

setDescriptions(String[]) Initialize the elements of an array of count descriptions.

setExitOnServerError(boolean) This property can only be set at design time.

setHost(String) This property can only be set at design time.

setInitialRetryDelay(int) This property can only be set at design time.

setMaxRetryDelay(int) This property can only be set at design time.

setPort(int) This property can only be set at design time.

setRetryDelayIncrement(int) This property can only be set at design time.

setRunServer(boolean) This property can only be set at design time.

setServerRetryDelay(long) This property can only be set at design time.

19.4.4 Fields

public static final int **CHANNEL_REQUESTS**

Index for count of channel requests.

public static final int **LOCAL_CHANNEL_ATTEMPT**

Index for count of local channel attempts.

public static final int **LOCAL_CHANNEL_REQUEST_SUCCESS**

Index for count of local channel attempts.

public static final int **LOCAL_CHANNEL_REQUEST_REJECT**

Index for count of local channel attempts.

public static final in**t LOCAL_CHANNEL_ACCEPT**

Index for count of local channel attempts.

public static final int **REMOTE_CHANNEL_ATTEMPT**

Index for count of remote channel attempts.

public static final int **REMOTE_CHANNEL_REQUEST_SUCCESS**

Index for count of local channel attempts.

public static final int **REMOTE_CHANNEL_REQUEST_REJECT**

Index for count of local channel attempts.

public static final int **REMOTE_CHANNEL_ACCEPT**

Index for count of local channel attempts.

public static final int **ABORTED_CHANNEL_REQUESTS**

Index for count of aborted channel requests.

public static final int **BYTES_READ**

Index for count of bytes read.

public static final int **BYTES_WRITTEN**

Index for count of bytes read.

public static final int **COUNT_SIZE**

Size of counts and limits arrays needed by Net.

19.4.5 Constructors

Net public Net()

19.4.6 Methods

public static final void setDescriptions(String d[])

Initialize the elements of an array of count descriptions.

public int getCacheDelay()

public void setCacheDelay(int v)

This property can only be set at design time.

public int getCacheExtent()

public void setCacheExtent(int v)

This property can only be set at design time.

public int getInitialRetryDelay()

public void setInitialRetryDelay(int v)

This property can only be set at design time.

public int getRetryDelayIncrement()

public void setRetryDelayIncrement(int v)

This property can only be set at design time.

public int getMaxRetryDelay()

public void setMaxRetryDelay(int v)

This property can only be set at design time.

public String getHost()

public void setHost(String v)

This property can only be set at design time.

public int getPort()

public void setPort(int v)

This property can only be set at design time.

public boolean getExitOnServerError()

public void setExitOnServerError(boolean flag)

This property can only be set at design time.

public boolean getRunServer()

public void setRunServer(boolean flag)

This property can only be set at design time.

public long getServerRetryDelay()

public void setServerRetryDelay(long delay)

This property can only be set at design time.

public void addChannelListener(CallBack listener) throws TooManyListenersException

Register for new channel notifications.

public void removeChannelListener(CallBack listener)

Unregister for new channel notifications.

public CallBack getRequestChannelCallBack()

Returns the CallBack used to pass channel requests.

public void processEvent(SandboxEvent se)

Not used.

public void processRequest(SandboxRequest channelRequest)

Sandbox requests are passed to net to request a channel.

public void requestTimeout(RequestControl rc)

public void init()

Extention of NetUser's init()

Overrides:

init in class EnvironmentChild

19.5 NetUser

```
java.lang.Object
   |
   +----ORG.opengroup.sandbox.environment.EnvironmentChild
           |
           +----ORG.opengroup.sandbox.net.NetUser
```

public class NetUser extends EnvironmentChild implements CallBackListener

19.5.1 Constructor Index

NetUser()

19.5.2 Method Index

getHost() getIncomingChannelListener()

Implemented by any component that uses Net.

getNetBean() Return the name of the net bean

getPort()

init() Initializations required by NetUser are done here

openChannel(String, String, int, byte, long) Opens a communication channel

processEvent(SandboxEvent) Implementation of the SandboxListener interface method.

processRequest(SandboxRequest) Implementation of the SandboxListener interface method.

requestTimeout(RequestControl)

setNetBean(String) Set the name of the net bean

19.5.3 Constructors

NetUser public NetUser()

19.5.4 Methods

setNetBean public void setNetBean(String netBean)

Set the name of the net bean

getNetBean public String getNetBean()

Return the name of the net bean

getIncomingChannelListener public CallBack getIncomingChannelListener()

Implemented by any component that uses Net. Returns the SandboxListener which is to receive incoming channel requests. This method allows the owner of Net to route Incoming Channel requests to the appropriate user.

init public void init()

Initializations required by NetUser are done here

Overrides:

init in class EnvironmentChild

processRequest public void processRequest(SandboxRequest sr)

Implementation of the SandboxListener interface method. Called for incoming channel requests. Must be overridden.

processEvent public void processEvent(SandboxEvent se)

Implementation of the SandboxListener interface method. (Not used)

requestTimeout public void requestTimeout(RequestControl rc)

openChannel public Object openChannel(String destName, String destHost, int destPort, byte att, long timeout)

Opens a communication channel

getHost public String getHost()

getPort public int getPort()

19.6 ChannelAccess (Interface)

public interface ChannelAccess

Limited access to a connection to another agent.

19.6.1 Method Indedx

close() Closes the connection.

getAgent() Returns the full name of the agent connected to.

getAttribute() Returns the attribute of the connection.

getHost() Returns the host name of the connection.

getInputStream() Returns the InputStream of the connection.

getOutputStream() Returns the OutputStream of the connection.

getPort() Returns the port of the connection.

setTimeout(int) Sets the connection timeout value.

19.6.2 Methods

public abstract String getHost() throws ClosedException

Returns the host name of the connection.

Throws: ClosedException

The connection was closed.

public abstract int getPort() throws ClosedException

Returns the port of the connection.

Throws: ClosedException

The connection was closed.

public abstract byte getAttribute() throws ClosedException

Returns the attribute of the connection.

Throws: ClosedException

The connection was closed.

public abstract String getAgent() throws ClosedException

Returns the full name of the agent connected to.

Throws: ClosedException

The connection was closed.

public abstract void setTimeout(int tmo) throws ClosedException

Sets the connection timeout value.

Parameters:

tmo - Timeout delay in milliseconds.

Throws: ClosedException

The connection was closed.

public abstract ObjectInputStream getInputStream() throws ClosedException

Returns the InputStream of the connection.

Throws: ClosedException

The connection was closed.

public abstract ObjectOutputStream getOutputStream() throws ClosedException

Returns the OutputStream of the connection.

Throws: ClosedException

The connection was closed.

public abstract void close()

Closes the connection.

20 moa.BatchDriver

This package contains the BatchDriver class which provides a script-based user interface to the MOA system. It provides users with various types of interactive, monitoring, and debugging services. It is primarily used for testing purposes. It is a component in the front end of the MOA system. It supports interaction with both the home-end and remote-end of the system. A user can log into the system from any remote location. BatchDriver object starts a thread which continuously reads from the input test file on a line by line basis till the end of file is reached. Each line represents a command. The first word of each file represents the command type (e.g. login, launch, etc.). The rest of the line consists of various arguments necessary for executing that command.

20.1 BatchDriver

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
       |
       +----ORG.opengroup.sandbox.message.MessageUser
            |
            +----ORG.opengroup.moa.batchdriver.BatchDriver
```

public class BatchDriver extends MessageUser implements Runnable

BatchDriver is a script-based interface enabling users to interactively monitor, and debug agents, to launch them, snapshot agents state, configure monitoring capabilities, and querying the logged data of an agent. It is usually used for testing purposes.

20.1.1 Constructor Index

BatchDriver()

20.1.2 Method Index

incomingRequest(String, String, int, byte, Object) This method handles the display of Movement, Notice, LogItem, and Stats objects received from the user

init() This method initializes the BatchDriver by creating a thread which continuously reads commands from the input script.

run()

This method reads the script line by line.

20.1.3 Constructors

BatchDriver public BatchDriver()

20.1.4 Methods

public void init()

This method initializes the BatchDriver by creating a thread which continuously reads commands from the input script. It is called by the parent root Environment during class loading

Overrides:

init in class MessageUser

public void run()

This method reads the script line by line. It parses the line to set the input command type and then calls the processCommand method to process that Command

public Object incomingRequest(String origName, String origHost, int origPort,byte att, Object content)

This method handles the display of Movement, Notice, LogItem, and Stats objects received from the user

Parameters:

origName - name of the originator of message

origHost - host of the originator of message

origPort - port of the originator of message

att - attribute of the message

content - the content of the message

Returns:

an Object indicating the result of the incomingRequest. If there was a failure a Failure object is returned

Overrides:

incomingRequest in class MessageUser

21 moa.MoaApplet

This package provides users with various types of interactive, monitoring, and debugging services. It provides an applet-based interface which is used by the user to interact with the MOA system.

The User Interface is a component in the front end of the system. It supports interaction with both the home-end and remote-end of the system. The MoaApplet class, when first instantiated brings up a Panel which enables the user to login. It uses the Communicator class which extend MessageUser to send and receive messages from the home-end and remote-end MOA systems. Upon a successful login it displays all the preconfigured and launched agents, and snapshots resident at the home node. The user can select any agent to send a command, or a query, or to display its family. The family can be modified by the user. All the incoming messages (e.g. Movement, Notice, LogItems etc.) are also displayed.

21.1 CommandDialog

```
java.lang.Object
  |
  +----java.awt.Component
         |
         +----java.awt.Container
                |
                +----java.awt.Window
                       |
                       +----java.awt.Dialog
                              |
                              +----ORG.opengroup.moa.moaApplet.CommandDialog
```

public class CommandDialog extends Dialog implements ActionListener

21.1.1 Constructor Index

CommandDialog(MoaApplet, String)

21.1.2 Method Index

actionPerformed(ActionEvent)

21.1.3 Constructors

CommandDialog public CommandDialog(MoaApplet f, String name)

21.1.4 Methods

public void actionPerformed(ActionEvent ae)

21.2 Communicator

```
java.lang.Object
  |
  +----ORG.opengroup.sandbox.environment.EnvironmentChild
         |
```

```
+----ORG.opengroup.sandbox.message.MessageUser
        |
        +----ORG.opengroup.moa.moaApplet.Communicator
```

public class Communicator extends MessageUser implements Runnable

This class provides the communication interface to the MoaApplet. It extends the MessageUser component and is thus able to send and receive messages. It has two queues: requestQueue and resultQueue. All the operator requests made from the MoaApplet are enqueued in the requestQueue. This class starts a special requestThread which dequeues the requests from the requestQueue, sends it to the User object, and enqueues the result in the resultQueue. The waiting MoaApplet thread receives the result as a result of dequeuing the resultQueue.

21.2.1 Constructor Index

Communicator()

21.2.2 Method Index

getUserHost() Returns the user host

getUserName() Returns the user name

getUserPort() Returns the user host

incomingRequest(String, String, int, byte, Object) This method handles the display of Movement, Notice, LogItem, and Stats objects received from the user

init() This method initializes the request and result queues, and starts the the thread which will continuously process the user requests

run() This method dequeues the requests from the requestQueue, sends it to the User object, and enqueues the result in the resultQueue.

setUserHost(String) Sets the host of the user

setUserName(String) Sets the user name to which messages are to be sent

setUserPort(int) Sets the port of the user

21.2.3 Constructors

Communicator public Communicator()

21.2.4 Methods

public void setUserName(String s)

Sets the user name to which messages are to be sent

public void setUserHost(String s)

Sets the host of the user

public void setUserPort(int s)

Sets the port of the user

public String getUserName()

Returns the user name

public String getUserHost()

Returns the user host

public int getUserPort()

Returns the user host

public void init()

This method initializes the request and result queues, and starts the the thread which will continuously process the user requests

Overrides:

init in class MessageUser

public Object incomingRequest(String origName, String origHost, int origPort, byte att, Object content)

This method handles the display of Movement, Notice, LogItem, and Stats objects received from the user

Parameters:

origName - name of the originator of message

origHost - host of the originator of message

origPort - port of the originator of message

att - attribute of the message

content - the content of the message

Returns:

an Object indicating the result of the incomingRequest. If there was a failure a Failure object is returned

Overrides:

incomingRequest in class MessageUser

public void run()

This method dequeues the requests from the requestQueue, sends it to the User object, and enqueues the result in the resultQueue.

21.3 CountDialog

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
```

```
+----java.awt.Window
        |
        +----java.awt.Dialog
                |
                +----ORG.opengroup.moa.moaApplet.CountDialog
```

public class CountDialog extends Dialog implements ActionListener

This class displays a dialog box which lists all counts with their descriptions. The user can add the counts to be monitored by the Family from this List. This class is instantiated from the FamilyInfoPanel when a add Counts request is made.

21.3.1 Constructor Index

CountDialog(FamilyInfoPanel, String[]) Creates a CountDialog

21.3.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons.

21.3.3 Constructors

CountDialog public CountDialog(FamilyInfoPanel f, String counts[]) Creates a CountDialog

Parameters:

f - the parent FamilyInfoPanel

counts - the complete list of counts from which the selection has to be done

21.3.4 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons. When "Ok" is clicked all the items in the count List are added to the Family

Parameters:

ae - the event which resulted in this method being called

21.4 CreatePCADialog

```
java.lang.Object
  |
  +----java.awt.Component
          |
          +----java.awt.Container
                  |
                  +----java.awt.Window
                          |
                          +----java.awt.Dialog
                                  |
                                  +----ORG.opengroup.moa.moaApplet.CreatePCADialog
```

public class CreatePCADialog extends Dialog implements ActionListener

This class displays a dialog box to input the name of the preconfigured agent to be created from the selected Sanpshot. It is instantiated from the SnapshotPanel, when the user selects a snapshot and clicks "CreatePCA" button.

21.4.1 Constructor Index

CreatePCADialog(SnapshotPanel) Constructs a CreatePCADialog

21.4.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons.

21.4.3 Constructors

CreatePCADialog public CreatePCADialog(SnapshotPanel sp)

Constructs a CreatePCADialog

Parameters:

sp - the SnapshotPanel from which this dialog box is created.

21.4.4 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons. When "Ok" is clicked the information from pca name field is collected to create a preconfigured agent

Parameters:

ae - the event which resulted in this method being called

21.5 ErrorMessageDialog

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Window
                           |
                           +----java.awt.Dialog
                                   |
                                   +----ORG.opengroup.moa.moaApplet.ErrorMessageDialog
```

public class ErrorMessageDialog extends Dialog implements ActionListener

This class creates a dialog box which displays an Error of any type as a result of operations in the moaApplet package

21.5.1 Constructor Index

ErrorMessageDialog(String, String) Constructs an ErrorMessageDialog

21.5.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent object from the "Ok" button.

21.5.3 Constructors

ErrorMessageDialog public ErrorMessageDialog(String label, String message)

Constructs an ErrorMessageDialog

21.5.4 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent object from the "Ok" button.

Parameters:

ae - the event which resulted in this method being called

21.6 FamilyInfoPanel

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Panel
                           |
                           +----ORG.opengroup.moa.moaApplet.FamilyInfoPanel
```

public class FamilyInfoPanel extends Panel implements ActionListener, ItemListener

This class displays the Family object. It shows whether the trace is on or off, the counts being monitored and the frequency at which they are being monitored, the watches, and the logtypes. The user is also allowed to turn the trace on/off, change the count frequency, add/remove counts, watches, and logTypes in order to modify the family.

21.6.1 Constructor Index

FamilyInfoPanel(MoaApplet) Constructs a FamilyInfoPanel

21.6.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Add" and "Remove" buttons.

itemStateChanged(ItemEvent)

This method is called as a result of generation of ItemEvent objects from the viewSelectionList object.

21.6.3 Constructors

FamilyInfoPanel public FamilyInfoPanel(MoaApplet f)

Constructs a FamilyInfoPanel

Parameters:

f - the parent MoaApplet container

21.6.4 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Add" and "Remove" buttons. When "Add" is clicked the dialog box to create the item selected in viewSelection (Watches/LogTypes/Counts) is displayed through which user can create and add these items to the family. When "Remove" is clicked the selected item is removed.

Parameters:

ae - the event which resulted in this method being called

public void itemStateChanged(ItemEvent ie)

This method is called as a result of generation of ItemEvent objects from the viewSelectionList object. The panel shows Watch/Count/LogType lists depending upon the selection

Parameters:

ie - the event which resulted in this method being called

21.7 HomeDialog

```
java.lang.Object
  |
  +----java.awt.Component
          |
          +----java.awt.Container
                  |
                  +----java.awt.Window
                          |
                          +----java.awt.Dialog
                                  |
                                  +----ORG.opengroup.moa.moaApplet.HomeDialog
```

public class HomeDialog extends Dialog implements ActionListener

This class creates a dialog box which displays the home and alternate home of the selected agent.

21.7.1 Constructor Index

HomeDialog(Hashtable, String) Constructs a HomeDialog

21.7.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Ok" buttons.

21.7.3 Constructors

HomeDialog public HomeDialog(Hashtable h, String filename)

Constructs a HomeDialog

Parameters:

h - the hashtable containing the AEname for home and altHome

flename - the name of the agent whose homes are to be displayed

21.7.4 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Ok" buttons. When "Ok" is clicked the dialog box is disposed.

Parameters:

ae - the event which resulted in this method being called

21.8 HostSelector

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Window
                           |
                           +----java.awt.Dialog
                                   |
                                   +----ORG.opengroup.moa.moaApplet.HostSelector
```

public class HostSelector extends Dialog implements ActionListener, ItemListener

21.8.1 Constructor Index

HostSelector(Frame, String, Vector)

21.8.2 Method Index

actionPerformed(ActionEvent)

itemStateChanged(ItemEvent)

main(String[])

parseList(Vector)

21.8.3 Constructors

HostSelector public HostSelector(Frame f, String title, Vector list)

21.8.4 Methods

public void parseList(Vector list)

public void actionPerformed(ActionEvent ae)

public void itemStateChanged(ItemEvent ie)

public static void main(String argv[])

21.9 LAPanel

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Panel
                           |
                           +----ORG.opengroup.moa.moaApplet.LAPanel
```

public class LAPanel extends Panel implements ActionListener

This class creates a panel which lists the launched agents for the user. User can select any of these agents and display its home, show and update its family, query its logger, list snapshots, or send it a command.

21.9.1 Method Index

actionPerformed(ActionEvent)

This method is called as a result of generation of ActionEvent objects from the "Query", "Command" and "ListSnapshots" button.

21.9.2 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Query", "Command" and "ListSnapshots" button. When "Query" is clicked the QueryDialog dialog box is created, when "Command" is clicked CommandDialog dialog box is created, and when "ListSnapshots" is clicked, SnapshotDialog box is created.

Parameters:

ae - the event which resulted in this method being called

21.10 LaunchDestinationDialog

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Window
                           |
                           +----java.awt.Dialog
                                   |
                                   +----ORG.opengroup.moa.moaApplet.LaunchDestinationDialog
```

public class LaunchDestinationDialog extends Dialog implements ActionListener

This class creates a dialog box to input the host and port of the destination at which the agent has to be launched

21.10.1 Constructor Index

LaunchDestinationDialog(MoaApplet, String) Creates the LaunchDestinationDialog

21.10.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons.

21.10.3 Constructors

LaunchDestinationDialog public LaunchDestinationDialog(MoaApplet f, String filename)

Creates the LaunchDestinationDialog

Parameters:

f - the parent MoaApplet for the applicatiin

filename - name of the file to be launched

21.10.4 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons. When "Ok" is clicked the preconfigured agent is launched to the destination host and port.

Parameters:

ae - the event which resulted in this method being called

21.11 LayoutConfigurator

```
java.lang.Object
   |
   +----ORG.opengroup.moa.moaApplet.LayoutConfigurator
```

public class LayoutConfigurator extends Object

This class is used by moaApplet package Containers to layout the components according to the GridBagLayout

21.11.1 Constructor Index

LayoutConfigurator(Container) Constructs LayoutConfigurator

21.11.2 Constructors

LayoutConfigurator public LayoutConfigurator(Container c)

Constructs LayoutConfigurator

Parameters:

c - the container in which the components have to be arranged

21.12 LogTypeDialog

```
java.lang.Object
  |
  +----java.awt.Component
          |
          +----java.awt.Container
                  |
                  +----java.awt.Window
                          |
                          +----java.awt.Dialog
                                  |
                                  +----ORG.opengroup.moa.moaApplet.LogTypeDialog
```

public class LogTypeDialog extends Dialog implements ItemListener, ActionListener

This class represents the dialog box to create a LogType. It is instantiated when an add LogType request is made from FamilyInfoPanel. User can set the kind, priority, and destination for the LogType from here.

21.12.1 Constructor Index

LogTypeDialog(FamilyInfoPanel) Creates a LogTypeDialog

21.12.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons.

itemStateChanged(ItemEvent) This method is called as a result of generation of ItemEvent objects from the kind Choice selector.

21.12.3 Constructors

LogTypeDialog public LogTypeDialog(FamilyInfoPanel f)

Creates a LogTypeDialog

Parameters:

f - the FamilyInfoPanel from which add Watch request was made

21.12.4 Methods

public void itemStateChanged(ItemEvent ie)

This method is called as a result of generation of ItemEvent objects from the kind Choice selector. When event type is of Channel/Message Dialog box displays input fields for host and port, but if Limit is the selected event type, input fields for Limit Index and Limit value are displayed

Parameters:

ie - the event which resulted in this method being called

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons. When "Ok" is clicked the information from different field is collected to create a LogType and added to the Family.

Parameters:

ae - the event which resulted in this method being called

21.13 LoginPanel

```
java.lang.Object
  |
  +----java.awt.Component
          |
          +----java.awt.Container
                  |
                  +----java.awt.Panel
                          |
                          +----ORG.opengroup.moa.moaApplet.LoginPanel
```

public class LoginPanel extends Panel implements ActionListener

This Panel is used by the logger to input username and password to log into MOA system.

21.13.1 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Login" button.

21.13.2 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Login" button. When "Login" is clicked the information from username and password field is collected send a Logon message to the user

Parameters:

ae - the event which resulted in this method being called

21.14 MoaApplet

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Panel
                           |
                           +----java.applet.Applet
                                   |
                                   +----ORG.opengroup.moa.moaApplet.MoaApplet
```

public class MoaApplet extends Applet implements ActionListener, ItemListener

MoaApplet is an applet-based interface enabling users to interactively monitor, and debug agents, to launch them, snapshot agents state, configure monitoring capabilities, and querying the logged data of an agent.

21.14.1 Constructor Index

MoaApplet()

21.14.2 Method Index

actionPerformed(ActionEvent) Implementation of ActionListener interface This method is called as a result of generation of ActionEvent objects from the "ShowHome", "ShowFamily" and "UpdateFamily" buttons.

getHostport() Returns the port number of the applet's host

getHosts() Returns the hosts to which an agent may go

getHosts(int) Returns the host at a given index

init() This method initializes the MoaApplet.

itemStateChanged(ItemEvent) This method is called as a result of generation of ItemEvent objects from the agent type Choice selector.

setHostport(int) sets the port number of the applet's host

setHosts(String, int) Sets the host to which an agent may go at a given index

setHosts(String[]) Sets the hosts to which an agent may go

21.14.3 Constructors

MoaApplet public MoaApplet()

21.14.4 Methods

public void setHostport(int i)

sets the port number of the applet's host

public int getHostport()

Returns the port number of the applet's host

public void setHosts(String h,int i)

Sets the host to which an agent may go at a given index

Parameters:

h - hosrname

i - the index

public void setHosts(String h[])

Sets the hosts to which an agent may go

public String getHosts(int i)

Returns the host at a given index

public String[] getHosts()

Returns the hosts to which an agent may go

public void init()

This method initializes the MoaApplet. It creates a LoginPanel so that user can log

Overrides:

init in class Applet

public void itemStateChanged(ItemEvent ie)

This method is called as a result of generation of ItemEvent objects from the agent type Choice selector. The agent display shows whichever agent type is selected (Preconfigured/Launched/Snapshot)

Parameters:

ie - the event which resulted in this method being called

public void actionPerformed(ActionEvent ae)

Implementation of ActionListener interface This method is called as a result of generation of ActionEvent objects from the "ShowHome", "ShowFamily" and "UpdateFamily" buttons. When "ShowHome" is clicked the HomeDialog dialog box is created, when "ShowFamily" is clicked, FamilyInfoPanel shows the Family of the selected agent, and when "UpdateFamily" is clicked, the selected agent's family is replaced by the displayed family

Parameters:

ae - the event which resulted in this method being called

21.15 PCAPanel

java.lang.Object

```
 |
 +----java.awt.Component
         |
         +----java.awt.Container
                 |
                 +----java.awt.Panel
                         |
                         +----ORG.opengroup.moa.moaApplet.PCAPanel
```

public class PCAPanel extends Panel implements ActionListener

This class creates a panel which lists the preconfigured agents for the user. User can select any of these, and display its home, show and update its family, and launch it to a remote location.

21.15.1 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Launch" button.

21.15.2 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Launch" button. When "Launch" is clicked the LaunchDestinationDialog dialog box is created, so that user can input launch destination

Parameters:

ae - the event which resulted in this method being called

21.16 QueryDialog

java.lang.Object

```
 |
 +----java.awt.Component
         |
         +----java.awt.Container
                 |
                 +----java.awt.Window
                         |
                         +----java.awt.Dialog
                                 |
                                 +----ORG.opengroup.moa.moaApplet.QueryDialog
```

public class QueryDialog extends Dialog implements ActionListener

This class creates a dialog box to create a LogQuery and send it to logger at either home or alternate home of the selected agent

21.16.1 Constructor Index

QueryDialog() Default constructor

QueryDialog(MoaApplet, String) Creates the QueryDialog

21.16.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "SendQuery" and "Cancel" buttons.

main(String[])

21.16.3 Constructors

QueryDialog public QueryDialog(MoaApplet ap, String agentName)

Creates the QueryDialog

Parameters:

f - the parent MoaApplet for the applicatiin

agentname - name of the agent to be queried

QueryDialog public QueryDialog()

Default constructor

21.16.4 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "SendQuery" and "Cancel" buttons. When "SendQuery" is clicked the logger at home or alternate home (as per selection) is queried.

Parameters:

ae - the event which resulted in this method being called

public static void main(String argv[])

21.17 SnapshotDialog

```
java.lang.Object
  |
  +----java.awt.Component
         |
         +----java.awt.Container
                |
                +----java.awt.Window
                       |
                       +----java.awt.Dialog
                              |
                              +----ORG.opengroup.moa.moaApplet.SnapshotDialog
```

public class SnapshotDialog extends Dialog implements ActionListener

This class creates a dialog box through which a user can list the snapshots of the selected launched agent at either its home or alternate home. The user can select one of the listed snapshots and fetch it.

21.17.1 Constructor Index

SnapshotDialog(MoaApplet, String) Creates the SnapshotDialog

SnapshotDialog(String)

21.17.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "ListSnapshots", "FetchSnapshot" and "Cancel" buttons.

main(String[])

21.17.3 Constructors

SnapshotDialog public SnapshotDialog(MoaApplet ap, String agentName)

Creates the SnapshotDialog

Parameters:

f - the parent MoaApplet for the application

agentname - name of the agent for which snapshots are to be listed

SnapshotDialog public SnapshotDialog(String agentName)

21.17.4 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "ListSnapshots", "FetchSnapshot" and "Cancel" buttons. When "ListSnapshots" is clicked the homeService at home or alternate home (as per selection) of the selected agent is asked for a SnapshotList. The result is a Vector of SnapshotIDs. After selecting a snapshot from the list, if "FetchSnapshot" is clicked, a request is sent to the HomeService to send back the Snapshot.

Parameters:

ae - the event which resulted in this method being called

public static void main(String argv[])

21.18 SnapshotPanel

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Panel
                           |
                           +----ORG.opengroup.moa.moaApplet.SnapshotPanel
```

public class SnapshotPanel extends Panel implements ActionListener

This class creates a panel which lists the snapshots which have been fetched. User can select any of these, and display its home, show and update its family, query it, or create a preconfigured agent from it.

21.18.1 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Query" and "CreatePCA" buttons.

21.18.2 Methods

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Query" and "CreatePCA" buttons. When "Query" is clicked the QueryDialog dialog box is created, and when "CreatePCA" is clicked CreatePCA dialog box is created.

Parameters:

ae - the event which resulted in this method being called

21.19 WatchDialog

```
java.lang.Object
   |
   +----java.awt.Component
           |
           +----java.awt.Container
                   |
                   +----java.awt.Window
                           |
                           +----java.awt.Dialog
                                   |
                                   +----ORG.opengroup.moa.moaApplet.WatchDialog
```

public class WatchDialog extends Dialog implements ItemListener, ActionListener

This class represents the dialog box to create a watch. It is instantiated when an add watch request is made from FamilyInfoPanel. User can set the event type, event, response, and Notice for the watch from here.

21.19.1 Constructor Index

WatchDialog(FamilyInfoPanel) Creates a WatchDialog

21.19.2 Method Index

actionPerformed(ActionEvent) This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons.

itemStateChanged(ItemEvent) This method is called as a result of generation of ItemEvent objects from the eventType Choice selector.

21.19.3 Constructors

WatchDialog public WatchDialog(FamilyInfoPanel f)

Creates a WatchDialog

Parameters:

f - the FamilyInfoPanel from which add Watch request was made

21.19.4 Methods

public void itemStateChanged(ItemEvent ie)

This method is called as a result of generation of ItemEvent objects from the eventType Choice selector. When event type is of Channel/Message Dialog box displays input fields for host and port, but if Limit is the selected event type, input fields for Limit Index and Limit value are displayed

Parameters:

ie - the event which resulted in this method being called

public void actionPerformed(ActionEvent ae)

This method is called as a result of generation of ActionEvent objects from the "Ok" and "Cancel" buttons. When "Ok" is clicked the information from different field is collected to create a watch and add to the Family.

Parameters:

ae - the event which resulted in this method being called