Date: 3/11/2002
Doc Version v.80
API Version 0.9
Authors: Chuck Shubert, Mark O’Neil

A. Executive Summary
The OKI Shared API is a collection of interfaces which are utilized in the Common Services Layer. These interfaces describe Agent, AgentIterator, Exception, Factory, FactoryManager, Group, and Type which interact with, and are utilized by the Common Services APIs. Examples of objects using the Agent interface are, a person, a requesting service, an external OKI instance, or an OKI tool. The Group interface extends Agent and allow the application to define a collection of Agents. The Factory interface provides the mechanism for the user of a Common Service API to access objects in the API. The FactoryManager class provides the mechanism to access implementation of the Factory interface. Type provides a class used in exchanging information between the application and the api implementation about the implementation and the details of the objects returned. Exception is the base class for OKI exceptions.

B. General Description
The Shared API interfaces fall into three categories; Those dealing with Agents, Factories, and Exceptions.

1. Agent
   getName() returns the name of the agent.
   getProxy() returns linked Agent

2. Group
   getName()  
   getAgents()  
   removeAgent()  
   addAgent()  

3. AgentIterator
   hasMoreAgents ()  
   nextAgent ()  

4. FactoryManager
   getDefaultFactory (  
   String apiPackageName,  
   String implPackageName,  
   Agent owner  
   )  
   Fetch the default OKI factory implementation for the requested package.
5. `getFactory`
   ```java
   getFactory(String apiPackageName,
               String implPackageName,
               Agent owner,
               java.util.Map props)
   ```
   Fetch a user defined factory implementation for the requested package.

6. Factory
   All factories implement this.
   ```java
   getOwner()
   setOwner(Agent Owner)
   getProperties()
   setProperties(java.util.Map properties)
   ```

7. Type
   General Type object, used in communicating between implementations of APIs and applications. APIs must define their own subclasses of Type. The domain specifies the context in which the Type is to be used. The keyword is used in distinguishing e.g. the type of authentication performed; the authority defines the namespace giving definition to the keyword, the optional authorityURI specifies where to find further information about the authority, and the optional info provides additional information about the type. Two Type objects are equal iff their authority and keyword strings are equal.
   ```java
   Type(java.lang.String domain,
        java.lang.String authorityURI,
        java.lang.String authority,
        java.lang.String keyword,
        java.lang.String info)
   ```

8. Exception
   Interface for OKI specific exceptions
   ```java
   UNIMPLEMENTED = "Unimplemented method ";
   DEFAULTFACTORYDIDNOTLOAD = "Default factory did not load ";
   FACTORYPROPERTIESDIDNOTLOAD = "Factory properties did not load ";
   ```
   ```java
   Exception(String message)
   ```

C. Technical Information

1. `org.okiproject.service.shared.api.Agent`
   Provides a serialized object wrapper for Agent type. Agent contains information about the agent; it's name, it's proxy.
   ```java
   public interface Agent extends java.io.Serializable {
       public java.lang.String getName();
       public Agent getProxy();
   }
   ```

2. `org.okiproject.service.shared.api.Agents`
   Provides a serialized object wrapper for Agent collections with Iterator methods.
   ```java
   public interface Agents extends java.io.Serializable {
       public boolean hasMoreAgents();
       public org.okiproject.service.shared.api.Agent
   ```
3. org.okiproject.service.shared.api.Group
Provides an object wrapper for authorizable Agent type Group.

```java
public interface Group extends Agent {
  public java.lang.String getName();
  public org.okip.service.shared.api.Agent[] getAgents();
  public void removeAgent();
  public void addAgent();
}
```

4. org.okiproject.service.shared.api.FactoryManager

```java
public interface FactoryManager implements java.io.Serializable {
  public static org.okip.service.shared.api.Factory getDefaultFactory(
    String apiPackageName,
    String implPackageName,
    org.okip.service.shared.api.Agent owner
  ); // Throws Exception
  public static org.okip.service.shared.api.Factory getFactory(
    String apiPackageName,
    String implPackageName,
    org.okip.service.shared.api.Agent owner,
    java.util.Map props
  ); // Throws Exception
}
```

5. org.okiproject.service.shared.api.Factory

```java
public interface Factory implements java.io.Serializable {
  public org.okiproject.service.shared.api.Agent getOwner();
  public void setOwner(org.okiproject.service.shared.api.Agent Owner);
  public java.util.Map getProperties();
  public void setProperties(java.util.Map properties);
}
```

6. org.okiproject.service.shared.api.Type

```java
public abstract class Type extends java.lang.Object implements java.io.Serializable {
  public boolean equals(Type type2)
  public java.lang.String getDomain()
  public java.lang.String getAuthority()
  public java.lang.String getAuthorityURI()
  public java.lang.String getKeyword()
  public char getSeparatorChar()
  public java.lang.String getInfo()
```
public void setSeparatorChar(char separatorChar)
public java.lang.String toString()
}

7. org.okiproject.service.shared.api.Exception
public class Exception extends java.lang.Exception {
    public static final String UNIMPLEMENTED = "Unimplemented method ";
    public static final String DEFAULTFACTORYDIDNOTLOAD = "Default factory did not load ";
    public static final String FACTORYPROPERTIESDIDNOTLOAD = "Factory properties did not load ";

    public Exception ( String message ) {
        super ( message );
    }
}