

## **Documentation for DirectoryObject.addProperty()**

Add/update the value of the specified property.

### **Synopsis**

*directoryObject.addProperty(name, value)*

Arguments

name

String representing the name of the property to be created/modified.

value

The value to add the property to.

### **Description**

This method adds the specified property if it does not already exist. This change is made locally to the script environment, not to the data store. The value can be a single value object or an array of objects. For multi valued objects, addProperty() will add the values to specified property in the directory object and NOT replace them. For single valued attribute, this property can only be used for adding the property if it doesn't exist but cannot be used to update it. The value's type (syntax) of object must be compatible with the syntax of the specified property. This method is available for the following data types:

- void addProperty(String name, Collection value);
- void addProperty(String name, Date value);
- void addProperty(String name, Map value);
- void addProperty(String name, boolean value);
- void addProperty(String name, byte value);
- void addProperty(String name, String value);
- void addProperty(String name, number value);

### **Usage**

```
directoryObject.addProperty("eruid", "jdoe");
```

The getProperty method returns a Java™ array of objects that is actually stored in a JavaScript™ JSONArray object. Unlike a standard JavaScript array, JSONArray objects are used to access members of a Java array. Since Java arrays cannot be resized, the size of a JSONArray object can not be changed. Also, JSONArray objects are typed. Setting a JSONArray element to the wrong type will throw a JavaScript error.

In Tivoli Identity Manager, a JSONArray object cannot be passed directly back into a addProperty method. The JSONArray array into a standard JavaScript array as follows:

```
jsAliases = new Array();
myPerson = person.get();
aliases = myPerson.getProperty("eralias");
for (i=0; i < aliases.length; i++) {
    jsAliases[i] = aliases[i];
}
jsAliases[aliases.length] = "myNewAlias";

myPerson.addProperty("eralias", jsAliases);
person.set(myPerson);
```