Creating Suite Definitions and Suite Terminologies

If a scriptable application is going to provide any scripting support beyond that defined by the Yellow Box frameworks or any other imported framework or loaded bundle, it must include a "suite definition" and, for each supported language, a "suite terminology."

A suite definition describes the scriptable objects of an application, framework, or bundle in terms of their attributes (properties), relationships (elements), and supported commands. A suite terminology describes the allowable AppleScript terms and phrases for each supported natural language. The attribute and relationship keys in the suite definition are mapped to the corresponding names in the suite terminology. The suite definition and the suite terminology also serve as the source of the summary information displayed by the Script Editor application.

Both sets of information are in files with distinct names and locations:

- Suite definitions are in files named *suiteName*.**scriptSuite** where *suiteName* is a name uniquely identifying the suite, and corresponding to the name of the suite terminology, if any. There should be only one file per suite. The file should be among the application's language-independent resources (that is, in the top level of the application wrapper's Resources directory).

- Suite terminologies are in files named *suiteName*.**scriptTerminology** where *suiteName* is a name uniquely identifying the suite, and corresponding to the name of the suite definition. A file for each localized language should be in the appropriate *language*.**lproj** directory of the application's resources.

The information in suite definitions and suite terminologies must be in a structured textual format known as a "property list." For the current release you must build these property lists "by hand"—that is, by using a text editor.

A Primer on ASCII Property Lists

In the Yellow Box, A property list is a structured representation of object data. It can be in binary or ASCII (that is, textual) format. The primary value of a property list is that it can be stored as an external source and read by a program at run time. Suite definitions and terminology definitions are property lists that represent the scriptable properties, elements, commands, and special terms of an application. They are loaded by an application when it is launched.

When an application loads an ASCII property list, the items in the property list are converted into Yellow Box objects based on how they are structured. These objects include NSString, NSData, NSArray, and NSDictionary objects. NSArrays and NSDictionaries are collection objects and thus can contain NSStrings and NSDatas as well as nested NSArrays and NSDictionaries.

A property list uses punctuation (curly braces, equal signs, parentheses, quotation marks, commas, and semicolons) to define its structure. Although indentation has no affect on how items are interpreted, property lists are frequently indented to improve readability. To represent NSDictionaries (as are suite terminologies and suite terminologies), you must enclose the entire property list in curly braces:

```
{
     // NSStrings, NSDatas, NSArrays,
     // and NSDictionaries represented here
}
```

The remainder of this section discusses how NSStrings, NSArrays, and NSDictionaries are represented in property lists. Because NSData objects are rarely represented in property lists—and do not occur in suite definitions or terminology definitions—they are not discussed here.

NSString

You represent an NSString by enclosing a symbol in quotation marks, for example:

```
“NSDocument”
```

(Sometimes quotes are optional, but it is recommended that you always use them.) The elements of an array are frequently NSStrings. The left (key) side of a key/value assignment used to define an item in a dictionary must be an NSString; the right (value) side *may* be an NSString. For example:

```
“Type” = “NSTextStorage”;
```

NSArray

You represent an NSArray by enclosing a comma-divided list with parentheses:

```
(“cha “, “ctxt”, “font”)
```

NSArrays are usually on the right (value) side of NSDictionary key/value assignments:

```
“SynonymAppleEventCodes” = (“cha “, “ctxt”, “font”);
```

(As noted earlier, you must always represent dictionary keys as NSStrings.) The elements of a represented array can be NSStrings, NSDatas, NSDictionaries, or other NSArrays. For example:

```
“TopLevelArray” = (“x”, (“a”, “b”, “c”), “y”, {“z” = “hello”});
```

NSDictionary

An item in an NSDictionary is a key/value pair represented as an assignment (that is, it uses an equals sign as a separator and terminates with a semicolon). For example:

```
“AppleEventCode” = “aevt”;
```

The left side of the assignment is referred to as the “key”; the key must be an NSString. The right side of the assignment is known as the “value”; it can be any allowable property-list type. For example, it could be another NSDictionary, which can itself nest other NSDictionaries:

```
{ // top-level dictionary
// other stuff...
“NSTextStorage” = {
     “Superclass” = “NSCoreSuite.AbstractObject”;
     “Attributes” = {
          “foregroundColor” = {
               “Type” = “NSColor”;
               “AppleEventCode” = “colr”;
          };
// more other stuff...
}
```

The Structure of a Suite Definition

The property list for a suite definition consists of a series of nested dictionaries. Use appropriate Apple event codes or, if none exists for a certain class of object, attribute, or relationship, use a four-letter Apple event code that you know to be unique.

Suite Dictionary

Key	Value Type or Reference	Description
“Name”	NSString	Name of suite (required)
“AppleEventCode”	NSString	Four-letter Apple event code for this suite (required)
“Classes”	Class List Dictionary	optional (no classes defined by default)
“Commands”	Command List Dictionary	optional (no classes defined by default)
“Synonyms”	Synonym List Dictionary	optional (no synonyms defined by default)

Class List Dictionary

Key	Reference	Description
“*className*”	Class Dictionary	One per each scriptable class. Must be the name of an Objective-C or Yellow Box Java class.

Class Dictionary

Key	Value Type or Reference	Description
“Superclass”	NSString	Scriptable superclass; must be the name of an Objective-C or Yellow Box Java class. All attributes, relationships, and supported commands are inherited and can be overriden. You can use the notation *suiteName.className* to designate the class. (Optional.)
“AppleEventCode”	NSString	Four-letter Apple event code for this class (required)
“Attributes”	Property List Dictionary	Attributes (properties) of the class (optional)
“ToOneRelationships”	Property List Dictionary	One-to-one relationships (elements) of the class (optional)
“ToManyRelationships”	Property List Dictionary	One-to-many relationships (elements) of the class (optional)
“SupportedCommands”	Supported Commands Dictionary	Commands supported by the class (optional)
“InverseRelationships”	Inverse Relationship Dictionary	An inverse relationship maps a given relationship to the relationship in the destination class of that relationship that points back to this class (optional).

Property List Dictionary

Key	Reference	Description
“*propertyName*”	Property Dictionary	Definition of attribute or relationship. *propertyName* should map to an instance variable of the class for which there are accessor methods.

Property Dictionary

Key	Value Type	Description
“Type”	NSString	Name of class for values of this property (required)
“AppleEventCode”	NSString	Four-letter Apple event code for this suite (required)
“ReadOnly”	NSString	“Yes” or “No” (“No” is default; optional)

Supported Commands Dictionary

Key	Value Type	Description
“*commandName*”	NSString	Name of method this class uses to implement the command or ““ if the default implementation is sufficient. *commandName* should be in *suiteName.commandName* notation if command is not in the same suite as the class.

Inverse Relationship Dictionary

Key	Value Type	Description
“*relationshipName*”	NSString	Name of an inverse relationship. An inverse relationship specifies a relationship to a class that points back to this class.

Command List Dictionary

Key	Reference	Description
“*commandName*”	Command Dictionary	Command definition.

Command Dictionary

Key	Value Type or Reference	Description
“CommandClass”	NSString	Class of command (optional; NSScriptCommand by default). If not default, must be a subclass of NSScriptCommand.
“AppleEventCode”	NSString	Four-letter Apple event code for this command (required)
“AppleEventClassCode”	NSString	Four-letter Apple event class code for this command (optional, is the Apple event code of the suite definition by default)
“Type”	NSString	Class name of result of command or ““ if no result (optional, no result by default)
“ResultAppleEventCode”	NSString	Four-letter Apple event code for the return type of the command. Must be present if “Type” value is assigned. Can be “****” if the return type is variable.
“Arguments”	Argument List Dictionary	Arguments of command (optional, no arguments by default)

Argument List Dictionary

Key	Value Type or Reference	Description
“Type”	NSString	Name of class for this argument (required)
“AppleEventCode”	NSString	Four-letter Apple event code for this argument (required)
“Optional”	NSString	“Yes” or “No” (optional, “No” by default)

Synonym List Dictionary

Key	Value Type or Reference	Description
“*Apple event code*”	NSString	Class name for which four-letter Apple event code is a synonym.

Suite Definition Example

The following example is the Core suite (**NSCoreSuite.scriptSuite**) provided by the Yellow Box frameworks. This file has probably changed since its inclusion in this document; you can obtain an updated version of this file in **/System/Library/Frameworks/Scripting.framework/Resources.**

The AbstractObject class specifies a base class that your scriptable classes can inherit from; the “Abstract” prefix indicates that an application should never export instances of the class.

```
{
    “Name” = “NSCoreSuite”;
    “AppleEventCode” = “core”;

    “Classes” = {
        “AbstractObject” = {
            “Attributes” = {
                “className” = {
                    “Type” = “NSString”;
                    “AppleEventCode” = “pcnm”;
                    “ReadOnly” = “YES”;
                };
                “classCode” = {
                    “Type” = “NSNumber”;
                    “AppleEventCode” = “pcls”;
                   “ReadOnly” = “YES”;
                };
            };
            “SupportedCommands” = {
                “NSCoreSuite.Get” = ““;
                “NSCoreSuite.Count” = ““;
                “NSCoreSuite.Exists” = ““;
                “NSCoreSuite.Move” = ““;
                “NSCoreSuite.Copy” = ““;
                “NSCoreSuite.Create” = ““;
                “NSCoreSuite.Delete” = ““;
                “NSCoreSuite.Set” = ““;
            };
            “AppleEventCode” = “cobj”;
        };
        “NSApplication” = {
            “Superclass” = “NSCoreSuite.AbstractObject”;
            “Attributes” = {
                “name” = {
                    “Type” = “NSString”;
                    “AppleEventCode” = “pnam”;
                     “ReadOnly” = “YES”;
                };
                “isActive” = {
                    “Type” = “NSNumber”;
                    “AppleEventCode” = “pisf”;
                     “ReadOnly” = “YES”;
                };
                “version” = {
                    “Type” = “NSNumber”;
                    “AppleEventCode” = “vers”;
                     “ReadOnly” = “YES”;
                };
            };
            “ToManyRelationships” = {
                “orderedDocuments” = {
                    “Type” = “NSDocument”;
                    “AppleEventCode” = “docu”;
                     “ReadOnly” = “YES”;
                };
                “orderedWindows” = {
                    “Type” = “NSWindow”;
                    “AppleEventCode” = “cwin”;
                     “ReadOnly” = “YES”;
                };
            };
            “AppleEventCode” = “capp”;
        };
        “NSDocument” = {
            “Superclass” = “NSCoreSuite.AbstractObject”;
            “Attributes” = {
                “lastComponentOfFileName” = {
                    “Type” = “NSString”;
                    “AppleEventCode” = “pnam”;
                };
                “fileName” = {
                    “Type” = “NSString”;
                “AppleEventCode” = “ppth”;
            };
            “isDocumentEdited” = {
                “Type” = “NSNumber”;
                “AppleEventCode” = “imod”;
                 “ReadOnly” = “YES”;
            };
        };
        “SupportedCommands” = {
            “NSCoreSuite.Print” = ““;
            “NSCoreSuite.Save” = “handleSaveScriptCommand:”;
            “NSCoreSuite.Close” = “handleCloseScriptCommand:”;
        };
        “AppleEventCode” = “docu”;
    };
    “NSWindow” = {
        “Superclass” = “NSCoreSuite.AbstractObject”;
        “Attributes” = {
            “hasCloseBox” = {
                “Type” = “NSNumber”;
                “AppleEventCode” = “hclb”;
                 “ReadOnly” = “YES”;
            };
            “hasTitleBar” = {
                “Type” = “NSNumber”;
                “AppleEventCode” = “ptit”;
                 “ReadOnly” = “YES”;
            };
            “orderedIndex” = {
                “Type” = “NSNumber”;
                “AppleEventCode” = “pidx”;
            };
            “isFloatingPanel” = {
                “Type” = “NSNumber”;
                “AppleEventCode” = “isfl”;
                 “ReadOnly” = “YES”;
            };
            “isModalPanel” = {
                “Type” = “NSNumber”;
                “AppleEventCode” = “pmod”;
                 “ReadOnly” = “YES”;
            };
            “isResizable” = {
                “Type” = “NSNumber”;
                “AppleEventCode” = “prsz”;
                 “ReadOnly” = “YES”;
            };
            “isZoomable” = {
                “Type” = “NSNumber”;
```

```
                  “AppleEventCode” = “iszm”;
                  “ReadOnly” = “YES”;
              };
              “isZoomed” = {
                  “Type” = “NSNumber”;
                  “AppleEventCode” = “pzum”;
              };
              “isMiniaturizable” = {
                  “Type” = “NSNumber”;
                  “AppleEventCode” = “ismn”;
                  “ReadOnly” = “YES”;
              };
              “isMiniaturized” = {
                  “Type” = “NSNumber”;
                  “AppleEventCode” = “pmin”;
              };
              “title” = {
                  “Type” = “NSString”;
                  “AppleEventCode” = “pnam”;
              };
              “isVisible” = {
                  “Type” = “NSNumber”;
                  “AppleEventCode” = “pvis”;
              };
          };
          “ToOneRelationships” = {
              “document” = {
                  “Type” = “NSDocument”;
                  “AppleEventCode” = “docu”;
                  “ReadOnly” = “YES”;
              };
          };
          “AppleEventCode” = “cwin”;
      };

“NSColor” = {
          “Superclass” = “NSCoreSuite.AbstractObject”;
          “AppleEventCode” = “colr”;
};
  };

  “Commands” = {
      “Get” = {
          “CommandClass” = “NSGetCommand”;
          “Type” = “NSObject”;
          “ResultAppleEventCode” = “****”;
          “AppleEventClassCode” = “core”;
          “AppleEventCode” = “getd”;
      };
```

```
“Set” = {
    “CommandClass” = “NSSetCommand”;
    “Type” = ““;
    “Arguments” = {
        “Value” = {
            “Type” = “NSObject”;
            “AppleEventCode” = “data”;
        };
    };
    “AppleEventClassCode” = “core”;
    “AppleEventCode” = “setd”;
};
“Count” = {
    “CommandClass” = “NSCountCommand”;
    “Type” = “NSObject”;
    “ResultAppleEventCode” = “****”;
    “AppleEventClassCode” = “core”;
    “AppleEventCode” = “cnte”;
};
“Exists” = {
    “CommandClass” = “NSExistsCommand”;
    “Type” = “NSObject”;
    “ResultAppleEventCode” = “****”;
    “AppleEventClassCode” = “core”;
    “AppleEventCode” = “doex”;
};
“Delete” = {
    “CommandClass” = “NSDeleteCommand”;
    “Type” = ““;
    “AppleEventClassCode” = “core”;
    “AppleEventCode” = “delo”;
};
“Move” = {
    “CommandClass” = “NSMoveCommand”;
    “Type” = ““;
    “Arguments” = {
        “ToLocation” = {
            “Type” = “NSPositionalReference”;
            “AppleEventCode” = “insh”;
        };
    };
    “AppleEventClassCode” = “core”;
    “AppleEventCode” = “move”;
};
“Copy” = {
    “CommandClass” = “NSCopyCommand”;
    “Type” = ““;
    “Arguments” = {
        “ToLocation” = {
                "Type" = "NSPositionalReference";
                "AppleEventCode" = "insh";
            };
        };
        "AppleEventClassCode" = "core";
        "AppleEventCode" = "clon";
    };
    "Create" = {
        "CommandClass" = "NSCreateCommand";
        "Type" = "NSObjectReference";
        "ResultAppleEventCode" = "obj ";
        "Arguments" = {
            "Location" = {
                "Type" = "NSPositionalReference";
                "AppleEventCode" = "insh";
            };
            "ObjectClass" = {
                "Type" = "NSNumber";
                "AppleEventCode" = "kocl";
            };
            "ObjectData" = {
                "Type" = "NSObject";
                "AppleEventCode" = "data";
                 "Optional" = "YES";
            };
            "KeyDictionary" = {
                "Type" = "NSDictionary";
                "AppleEventCode" = "prdt";
                 "Optional" = "YES";
            };
        };
        "AppleEventClassCode" = "core";
        "AppleEventCode" = "crel";
    };
    "Save" = {
        "CommandClass" = "NSScriptCommand";
        "Type" = "";
        "Arguments" = {
            "File" = {
                "Type" = "NSString";
                "AppleEventCode" = "kfil";
                 "Optional" = "YES";
            };
            "FileType" = {
                "Type" = "NSString";
                "AppleEventCode" = "fltp";
                 "Optional" = "YES";
            };
        };
```

```
                    “AppleEventClassCode” = “core”;
                    “AppleEventCode” = “save”;
                };
                “Close” = {
                    “CommandClass” = “NSScriptCommand”;
                    “Type” = ““;
                    “Arguments” = {
                        “File” = {
                            “Type” = “NSString”;
                            “AppleEventCode” = “kfil”;
                            “Optional” = “YES”;
                        };
                        “SaveOptions” = {
                            “Type” = “NSNumber”;
                            “AppleEventCode” = “savo”;
                            “Optional” = “YES”;
                        };
                    };
                    “AppleEventClassCode” = “core”;
                    “AppleEventCode” = “clos”;
                };
                “Open” = {
                    “CommandClass” = “NSScriptCommand”;
                    “Type” = ““;
                    “AppleEventClassCode” = “core”;
                    “AppleEventCode” = “odoc”;
                };
                “Print” = {
                    “CommandClass” = “NSScriptCommand”;
                    “Type” = ““;
                    “AppleEventClassCode” = “core”;
                    “AppleEventCode” = “pdoc”;
                };
            };
        }
```

The Structure of a Suite Terminology

The property list for a suite terminology also consists of a series of nested dictionaries. Many of the subdictionaries (class, command, argument, and so on) should have counterparts in the suite definition.

Terminology Dictionary

Key	Value Type or Reference	Description
“Name”	NSString or NSArray of NSStrings	Human-readable name of suite (required)
“Description”	NSString	Human-readable description of suite (optional)
“Classes”	Class List Terminology Dictionary	Required only if there is a corresponding definition
“Commands”	Command List Terminology Dictionary	Required only if there is a corresponding definition
“Synonyms”	Class Synonym List Terminology Dictionary	Required only if there is a corresponding definition

Class List Terminology Dictionary

Key	Reference	Description
“*className*”	Class Terminology Dictionary	One per each scriptable class. Must be the name of an Objective-C or Yellow Box Java class.

Class Terminology Dictionary

Key	Value Type or Reference	Description
“Name”	NSString or NSArray of NSStrings	Human-readable name of class (required)
“Description”	NSString	Human-readable description of class (optional)
“PluralName”	NSString	Human-readable name for plural form of class (required)
“Attributes”	Property List Terminology Dictionary	Attributes (properties) of the class (required only if there is a corresponding definition)

Property List Terminology Dictionary

Key	Reference	Description
“*propertyName*”	Property Terminology Dictionary	Description of attribute (property) or relationship (element) of the class

Property Terminology Dictionary

Key	Value Type or Reference	Description
“Name”	NSString or NSArray of NSStrings	Human-readable name of property (required).
“Description”	NSString	Human readable description of property (optional)

Key	Value Type or Reference	Description
“Sex”	NSString or NSArray of NSStrings	“masculine”, “feminine”, or “none” (default).
“Number”	NSString or NSArray of NSStrings	“plural” or “singular” (default).

Command List Terminology Dictionary

Key	Reference	Description
"*className*"	Command Terminology Dictionary	Description of command supported by class.

Command Terminology Dictionary

Key	Value Type or Reference	Description
"Name"	NSString or NSArray of NSStrings	Human-readable name of command (required)
"Description"	NSString	Human-readable description of command (optional)
"IsVerb"	NSString or NSArray of NSStrings	"No" or "Yes" (default)
"Arguments"	Argument List Terminology Dictionary	Description of command arguments (required only if there is a definition)

Argument List Terminology Dictionary

Key	Reference	Description
"*argumentName*"	Agument Terminology Dictionary	Descriptions of command arguments.

Agument Terminology Dictionary

Key	Value Type or Reference	Description
"Name"	NSString or NSArray of NSStrings	Human-readable name of argument (required)
"Description"	NSString	Human-readable description of argument (optional)
"Sex"	NSString or NSArray of NSStrings	"masculine", "feminine", "none" (default)
"Number"	NSString or NSArray of NSStrings	"plural" or "singular" (default)

Class Synonym List Terminology Dictionary

Key	Reference	Description
"*Apple event code*"	Class Synonym Terminology Dictionary	Descriptions of Apple event code synonyms for class

Class Synonym Terminology Dictionary

Key	Value Type or Reference	Description
“Name”	NSString or NSArray of NSStrings	Human-readable name of class (required)
“Description”	NSString	Human-readable description of class (optional)
“PluralName”	NSString	Human-readable name of plural form of class (required)

Example of Suite Terminology

The following example is the Core suite (**NSCoreSuite.scriptTerminology**) provided by the Yellow Box frameworks. This file has probably changed since its inclusion in this document; you can obtain an updated version of this file in **/System/Library/Frameworks/Scripting.framework/Resources/English.lproj.**

```
{
    “Name” = “Standard Suite”;
    “Description” = “Common classes and commands for most
applications.”;
    “Classes” = {
         “AbstractObject” = {
             “Name” = “Abstract object”;
             “Description” = “Abstract object provides a base class for
scripting classes.  It is never used directly.”;
             “Attributes” = {
                 “className” = {
                       “Name” = “classname”;
                       “Description” = “The name of the class of the
object.”;
                  };
                  “classCode” = {
                       “Name” = “class”;
                       “Description” = “The class of the object.”;
                  };
             };
         };
         “NSApplication” = {
             “Name” = “application”;
             “Description” = “An application’s top level scripting
object.”;
            “PluralName” = “applications”;
            “Attributes” = {
                  “name” = {
                       “Name” = “name”;
                       “Description” = “The name of the application.”;
                  };
                  “isActive” = {
                       “Name” = “frontmost”;
```

```
                        “Description” = “Is this the frontmost (active) 
application?”;
                   };
                   “version” = {
                        “Name” = “version”;
                        “Description” = “The version of the application.”;
                   };
             };
           };
           “NSDocument” = {
               “Name” = “document”;
               “Description” = “A document.”;
              “PluralName” = “documents”;
               “Attributes” = {
                    “lastComponentOfFileName” = {
                         “Name” = “name”;
                         “Description” = “The document’s name.”;
                    };
                    “fileName” = {
                         “Name” = “path”;
                         “Description” = “The document’s path.”;
                    };
                    “isDocumentEdited” = {
                         “Name” = “modified”;
                         “Description” = “Has the document been modified 
since the last save?”;
                    };
               };
           };
           “NSWindow” = {
               “Name” = “window”;
               “Description” = “A window.”;
               “PluralName” = “windows”;
               “Attributes” = {
                   “hasCloseBox” = {
                         “Name” = “has close box”;
                         “Description” = “Whether the window has a close 
box.”;
                    };
                    “hasTitleBar” = {
                         “Name” = “has title bar”;
                         “Description” = “Whether the window has a title 
bar.”;
                    };
                    “orderedIndex” = {
                         “Name” = “index”;
                         “Description” = “The index of the window in the 
back-to-front window ordering.”;
                    };
                    “isFloatingPanel” = {
```

```
                              “Name” = “is floating”;
                              “Description” = “Whether the window floats.”;
                         };
                         “isModalPanel” = {
                             “Name” = “is running modal”;
                              “Description” = “Whether the window is the
application’s current modal window.”;
                         };
                         “isResizable” = {
                              “Name” = “is resizable”;
                              “Description” = “Whether the window can be
resized.”;
                         };
                         “isZoomable” = {
                              “Name” = “is zoomable”;
                              “Description” = “Whether the window can be
zoomed.”;
                         };
                         “isZoomed” = {
                              “Name” = “is zoomed”;
                              “Description” = “Whether the window is currently
zoomed.”;
                         };
                         “isMiniaturizable” = {
                              “Name” = “is miniaturizable”;
                              “Description” = “Whether the window can be
miniaturized.”;
                         };
                         “isMiniaturized” = {
                              “Name” = “is miniaturized”;
                              “Description” = “Whether the window is currently
miniaturized.”;
                         };
                         “title” = {
                              “Name” = “name”;
                              “Description” = “The full title of the window.”;
                         };
                         “isVisible” = {
                              “Name” = “is visible”;
                              “Description” = “Whether the window is currently
visible.”;
                         };
                    };
               };
               “NSColor” = {
                    “Name” = “color”;
                    “Description” = “A color.”;
                    “PluralName” = “colors”;
               };
          };
```

```
    “Commands” = {
         “Get” = {
            “Name” = “get”;
            “Description” = “Get the data for an object.”;
  };
         “Set” = {
            “Name” = “set”;
            “Description” = “Set an object’s data.”;
            “Arguments” = {
                “Value” = {
                  “Name” = “to”;
                  “Description” = “The new value.”;
                 };
           };
      };
         “Count” = {
            “Name” = “count”;
            “Description” = “Return the number of elements of a particular
class within an object.”;
         };
         “Exists” = {
            “Name” = “exists”;
            “Description” = “Verify if an object exists.”;
         };
         “Delete” = {
            “Name” = “delete”;
            “Description” = “Delete an object.”;
         };
         “Move” = {
            “Name” = “move”;
            “Description” = “Move object(s) to a new location.”;
            “Arguments” = {
                “ToLocation” = {
                  “Name” = “to”;
                  “Description” = “The new location for the object(s).”;
                 };
              };
          };
         “Copy” = {
            “Name” = “duplicate”;
            “Description” = “Copy object(s) and put the copies at a new
location.”;
            “Arguments” = {
                “ToLocation” = {
                  “Name” = “to”;
                  “Description” = “The location for the new object(s).”;
                 };
                };
            };
           “Create” = {
               “Name” = “make”;
               “Description” = “Make a new object.”;
               “Arguments” = {
                   “Location” = {
                         “Name” = “at”;
                         “Description” = “The location at which to insert
the object.”;
              };
                   “ObjectClass” = {
                         “Name” = “new”;
                         “Description” = “The class of the new object.”;
                   };
                   “ObjectData” = {
                         “Name” = “with data”;
                         “Description” = “The initial data for the
object.”;
                   };
                   “KeyDictionary” = {
                         “Name” = “with properties”;
                         “Description” = “The initial values for
properties of the object.”;
                   };
                };
            };
           “Save” = {
               “Name” = “save”;
               “Description” = “Save an object.”;
               “Arguments” = {
                   “File” = {
                       “Name” = “in”;
                       “Description” = “The file in which to save the
object.”;
                   };
                   “FileType” = {
                       “Name” = “as”;
                       “Description” = “The file type in which to save
the data.”;
                   };
               };
           };
           “Close” = {
               “Name” = “close”;
               “Description” = “Close an object.”;
               “Arguments” = {
                   “File” = {
                       “Name” = “saving in”;
```

```
                    “Description” = “The file in which to save the
object.”;
                };
                “SaveOptions” = {
                    “Name” = “saving”;
                    “Description” = “Specifies whether changes should be
saved before closing.”;
                }};
            };
         };
         “Open” = {
            “Name” = “open”;
            “Description” = “Open an object.”;
         };
         “Print” = {
            “Name” = “print”;
            “Description” = “Print an object.”;
         };
    };
}
```