The C Routines of the Omegahat R-Java Interface

Duncan Temple Lang
John Chambers

July 26, 2000

1 Stand-alone C Routines for Java

Table 1 provides a description of some of the basic routines and macros.

`MUST_GET_METHOD_ID()`	locates the Java method identifier given a class, method name and signature.
`get_method_id()`	find the handle for the Java method identified by name and signature in one of the specified classes.
`get_object_class_name()`	invoke the equivalent of the Java expression `obj.getClass().getName()` for the specified object.
`get_java_class_name()`	invokes the `getName()` on the specified class.
`RS_JAVA(getSystemProperty)()`	C version of `System.getProperty(name)`
`create_Java_vm()`	start the JVM.
`RS_JAVA(terminateJava)()`	shutdown the JVM.
`getJNIEnv()`	get the default JNI environment (*JNIEnv*) object.
`getThreadJNIEnv()`	get the JNI environment (*JNIEnv*) object for the thread in which this C code is executing.
`releaseThreadJNIEnv()`	detach the JNI environment for this thread.
`getJavaVM()`	get the global Java Virtual Machine pointer.
`exists_Java_vm()`	has the JVM been initialized and is it still running.

Initialization

`initReferenceClasses()`	caches references to the basic reference classes used in the interface to exchange objects to and from R and Omegahat.
`initializeOmegahatManagerMethods()`	cache methods used to invoke calls to the *OmegaInterfaceManager* such as the method, constructor execution, etc.
`VMNewStringArray()`	create an (empty) array of strings.

Exceptions

`RSJava(makeExceptionMessage)()`	concat the specified string with a general message announcing a Java error/exception.
`get_Java_exception()`	Determines whether an exception has occurred and is pending and clears it if there is one, returning the error message associated with that, generated from `get_Java_exception_message()`.
`java_method_exception()`	create a message indicating that a method of the specified name could not found.
`get_Java_exception_class()`	compute the class of an exception
`get_Java_exception_message()`	extract the message from a Java exception and create a version that has the general announcement about a Java exception/error.
`throw_Java_exception()`	throw a java exception of the specified class (or by default Exception) with the given message. This is equivalent to `throw new Exception(msg)` or some other class of exception.
`clear_Java_exception()`	removes and discards the pending exception.

Invocation via Omegahat

`callConstructorMethod()`	have Omegahat invoke a constructor, given the classname and array of arguments and other information.
`callGenericOmegaMethod()`	have Omegahat invoke a Java method given the target object or class, method name, arguments and other information such as method signature, etc.
`evalOmegaExpression()`	have the Omegahat interpreter evaluate the specified string, optionally given a collection of named objects to substitute into the expression.
`getInterfaceManager()`	retrieve the single *OmegaInterfaceManager* which is the Omegahat interpreter, evaluator and reference manager.

MetaForeignReferenceClass	C-level handle for the Java class *org.omegahat.Interfaces.NativeInterface.MetaForeignReference*
NullObject	The Java **null** object.
OmegaAnonymousReferenceClass	C-level handle for the Java class *org.omegahat.Interfaces.NativeInterface.AnonymousReference*
OmegaNamedReferenceClass	C-level handle for the Java class *org.omegahat.Interfaces.NativeInterface.NamedReference*
OmegahatInterfaceManagerClass	C-level handle for the class of the Omegahat interpreter/evaluator. (The actual class can be specified by the user.)
ReferenceClassNameMethodID	the `getClassName()` identifier for the *InterfaceReference* class.
ReferenceNameMethodID	handle for the `key()` method of the class *InterfaceReference* class.

Table 2: Constants & Global Variables

2 Routines From Within R

`IS()`
equivalent to the R function *inherits()*, checking whether an R object's list of classes contains a specific class name (string).

`JavaObjectArray()`
create an empty Java array of objects.

`JavaStringArray()`
create an empty Java array of String objects.

`ManagerFindClass()`
Calls the Omegahat evaluator's `findClass()` method to retrieve a `Class` object using the different class lists it manages (i.e. including dynamically generated classes, locally added classpath elements, etc.) and using partial name matching.

`RJava_createCall()`
internal routine that creates a R function call object that can then be evaluated. It takes a list of arguments and the function object that is to be called.

`RJava_defaultHandlerFunction()`
returns the R object (not a copy of it) that is registered as the foreign reference manager that brokers the calls to functions on the foreign references.

`RJava_setDefaultHandlerFunction()`
sets the object that is to be considered the foreign reference manager and to which method requests on the foreign references should be sent.

`RJava_invokeRReferenceMethod()`
passes the given list of arguments to the currently registered foreign reference manager which then extracts the reference identifier, method name and arguments and calls that function.

`RS_ConstructorJavaCall()`
R/S entry point called from *.JavaConstructor()* to dispatch a call to the Omeghat interpreter to create a Java object, passing the class name or reference to the object and a list of arguments.

`RS_JAVA(MethodConverter)()`
converter routine that translates a Java `java.lang.reflect.Method` or `java.lang.reflect.Constructor` to an R object.

`RS_JAVA(PropertyConverter)()`
this converts a Java `Properties` object into an R named character vector.

`RS_JAVA(RealVariableConverter)()`
an example of how to write a C-level converter from Java to R, this one converting an Omegahat *RealVariable* to an R numeric vector.

`RS_JAVA(ReferenceClassMatch)()`
a built-in converter matching routine that returns true if the specified Java object is an Omegahat reference object.

`RS_JAVA(ReferenceConverter)()`
converts an Omegahat reference object (to a Java object stored in one of the Anonymous or Named databases) to an R object.

`RS_JAVA(functionConverterMatch)()`
calls the R function registered to determine whether the other function registered with it is capable of converting the particular Java object.

`RS_JAVA(RfunctionConverter)()`
calls the R function registered to perform conversion from Java to R.

`RS_JAVA(callRConverterFunction)()`
routine called by the previous two to invoke the call to the R function, giving it the java object reference and class name.

`RS_JAVA(addStringValue)()`
inserts an R character vector of length 1 created by copying a Java `String` into a specified element of an R list.

`RS_JAVA(addStringValues)()`
inserts an R character vector created by copying a Java array of `String`s into a specified element of an R list.

`RS_JAVA(isJVMInitialized)()`
query whether the JVM has been started and is still running.

`RS_OmegahatExpression()`
R/S entry point called from *.OmegahatExpression()* to evaluate an Omegahat expression represented as a

FOREIGN˙REFERENCE˙CLASSNAME˙SLOT	index of the `className` field of an R foreign reference object
FOREIGN˙REFERENCE˙CLASS˙SLOT	index of the target classes field of an R foreign reference object
FOREIGN˙REFERENCE˙ID˙SLOT	index of the name or `key` field of an R foreign reference object
ManagerAssignMethodID	Java method handle for the `assign()` method which puts a Java object into an Omegahat database, either anonymous or named depending on the arguments.
FromJavaConverters	the linked list of elements for converting from Java objects to R objects.
ToJavaConverters	the linked list of elements for converting from R objects to Java objects.
BaseOmegahatReferenceClass	name of the R class representing a generic (i.e. neither anonymous or named) Omegahat reference object
AnonymousOmegahatReferenceClass	name of the R class representing an anonymous Omegahat reference
AnonymousRReferenceClass	name of the R class representing an anonymous R foreign reference C-level class identifier for the Java `AnonymousRReference`
NamedOmegahatReferenceClass	name of the R class representing an named Omegahat reference
NamedRReferenceClass	name of the R class representing a named (non-anonymous) R foreign reference

Table 3: R-Java Constants

`Java_org_omegahat_R_Java_RForeignReference_eval()`	the primary routine that performs the callbacks to R functions. This is usually called by the methods of classes dynamically generated via the *ForeignClassInterfaceGenerator* to implement a Java interface using an R object and its functions.
`Java_org_omegahat_R_Java_RManualFunctionActionListener_actionPerformed()`	a manually created example of calling an R function to implement a Java method.
`getREventCommand()`	

Table 4: JNI Native Methods