

# The C Routines of the Omegahat R-Java Interface

Duncan Temple Lang  
John Chambers

December 13, 2005

## 1 Stand-alone C Routines for Java

Table 2 provides a description of some of the basic routines and macros.

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

#### Initialization

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

#### Exceptions

<code>RSJava(makeExceptionMessage)()</code>	concat the specified string with a general message announcing a Java error/exception.
<code>get_Java_exception()</code>	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 <code>get_Java_exception_message()</code> .
<code>java_method_exception()</code>	create a message indicating that a method of the specified name could not found.
<code>get_Java_exception_class()</code>	compute the class of an exception
<code>get_Java_exception_message()</code>	extract the message from a Java exception and create a version that has the general announcement about a Java exception/error.
<code>throw_Java_exception()</code>	throw a java exception of the specified class (or by default <code>Exception</code> ) with the given message. This is equivalent to <code>throw new Exception(msg)</code> or some other class of exception.
<code>clear_Java_exception()</code>	removes and discards the pending exception.

#### Invocation via Omegahat

<code>callConstructorMethod()</code>	have Omegahat invoke a constructor, given the classname and array of arguments and other information.
<code>callGenericOmegaMethod()</code>	have Omegahat invoke a Java method given the target object or class, method name, arguments and other information such as method signature, etc.
<code>evalOmegaExpression()</code>	have the Omegahat interpreter evaluate the specified string, optionally given a collection of named objects to substitute into the expression.

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

Table 3: Constants &amp; Global Variables

## 2 Routines From Within R

IS()	equivalent to the R function <i>inherits()</i> , 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 <code>findClass()</code> method to retrieve a <code>Class</code> 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 <i>.JavaConstructor()</i> to dispatch a call to the Omegahat 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 <code>java.lang.reflect.Method</code> or <code>java.lang.reflect.Constructor</code> to an R object.
RS_JAVA(PropertyConverter)()	this converts a Java <code>Properties</code> 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 <i>RealVariable</i> 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 <code>String</code> into a specified element of an R list.
RS_JAVA(addStringValues)()	inserts an R character vector created by copying a Java array of <code>Strings</code> into a specified element of an R list.
RS_JAVA(isJVMInitialized)()	query whether the JVM has been started and is still running.

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

Table 4: R-Java Constants

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

Table 5: JNI Native Methods