Chapter 1. Admin Guide

Table of Contents

Overview

This module allows executing Java compiled classes from config file, exporting functions to access the SIP message from Java using Java Native Interface (JNI).

Dependencies

Kamailio Modules

The following modules must be loaded before this module:

- *none*.

External Libraries or Applications

The following packages are runtime libraries, required to launch

- *java-common* Base of all Java packages.
- *default-jre* Standard Java or Java compatible Runtime.
- *gcj-jre* Java runtime environment using GIJ/classpath.
- *libgcj12 (>=12)* Java runtime library for use with gcj.

The following packages are optional, required for development

- *ant* Java based build tool like make.
- *ant-contrib* Collection of tasks, types and other tools for Apache Ant.
- *ant-gcj* Java based build tool like make (GCJ).

- *default-jdk* Standard Java or Java compatible Development Kit
- *gcj-jdk* gcj and classpath development tools for Java(TM)
- *libgcj13-dev (>=12)* Java development headers for use with gcj
- *jdk* JDK Development Kit (either oracle jdk or openjdk)

The following libraries or applications must be compiled before running Kamailio with this module loaded:

The following packages are runtime libraries, required to launch

- <*class_name*>.class
- *kamailio*.jar

Java runtime

Java runtime library (JRE or JDK) is required to use this module.

Parameters

`class_name` (string)

The class name should have the same compiled file name. If the value is *"Kamailio"*, then the compiled file should be named as *"Kamailio.class"*.

Default value is “Kamailio”.

Example 1.1. Set `class_name` parameter

```
...
modparam("app_java", "class_name", "Kamailio")
...
```

`child_init_method` (string)

TBD.

Default value is “child_init”.

Example 1.2. Set `child_init_method` parameter

```
...
modparam("app_java", "child_init_method", "my_mod_init")
...
```

`java_options` (string)

Java options for Java Virtual Machine. For more info read *java docs* [http://docs.oracle.com/javase/6/docs/technotes/tools/windows/java.html]

Default value is “-Djava.compiler=NONE”.

Example 1.3. Set `java_options` parameter

```
...
modparam("app_java", "java_options", "-Djava.compiler=NONE")
...
```

Example 1.4. Set `java_options` parameter (live configuration)

```
...
# Assumes "application java folder" is located at /opt/kamailio/java
modparam("app_java", "java_options", "-Djava.compiler=NONE -Djava.class.path=/pa
...
```

Example 1.5. Set `java_options` parameter (verbose configuration)

```
...
# Assumes "application java folder" is located at /opt/kamailio/java
modparam("app_java", "java_options", "-verbose:gc,class,jni -Djava.compiler=NONE
...
```

Example 1.6. Set `java_options` parameter (debug configuration)

```
...
# Assumes "application java folder" is located at /opt/kamailio/java
modparam("app_java", "java_options", "-Xdebug -verbose:gc,class,jni -Djava.compi
...
```

`force_kam_cmd_exec` (int)

This parameter forces execution a kamailio comnmand with java native method “KamExec”. # Note: this is an untested yet feature, may cause (but may not) a memory leaks if used from embedded languages.

Default value is “0 (off)”.

Example 1.7. Set `force_kam_cmd_exec` parameter

```
...
modparam("app_java", "force_kam_cmd_exec", 1)
...
```

Functions

Common requirements

Each function has a required parameter “method_signature”. For more info see *Determine the signature of a method* [http://www.rgagnon.com/javadetails/java-0286.html]. Signature represents the variable type. The mapping between the Java type and C type is

```
Type     Chararacter
boolean      Z
byte         B
char         C
double       D
float        F
```

```
int          I
long         J
object       L
short        S
void         V
Note that to specify an object, the "L" is followed by the object's class nam
```

app_java supports the following signatures:

```
Primitives: Z,B,C,D,F,I,J,L,S,V
Objects:
 Ljava/lang/Boolean;
 Ljava/lang/Byte;
 Ljava/lang/Character;
 Ljava/lang/Double;
 Ljava/lang/Float;
 Ljava/lang/Integer;
 Ljava/lang/Long;
 Ljava/lang/Short;
 Ljava/lang/String;
 NULL parameter: V

Each parameter passed to function will be cast according to given signature.

Parameters are optional, ommitting a parameter meant the passed value is NULL.
Parameters count should be exactly the same as signature count.
Note 1: Arrays representation (symbol '[') is not supported yet.
Note 2: You shall use a correct signature, e.g. the following examples of combi
    java_method_exec("ExampleMethod", "ZI", "False");
       java_method_exec("ExampleMethod", "LI", "something", "5");
```

java_method_exec(method, method_signature, [param1[, param2[, ...]]])

Executes a java class method *method*. Parameter *method_signature* is required.

- **Example 1.8. Signature: "V"**

  Kamailio prototype

  ```
  java_method_exec("ExampleMethod", "V");
  ```

  Java prototype

  ```
  public int ExampleMethod();
  ```

  Example of usage:

  ```
  # Kamailio
  java_method_exec("ExampleMethod", "V");

  # Java
  public int ExampleMethod()
  {
    ... do something;
    return 1;
  ```

```
}
```

- **Example 1.9. Signature: "Ljava/lang/String;I"**

  Kamailio prototype

  ```
  java_method_exec("ExampleMethod", "Ljava/lang/String;I", "Hello world", "5");
  ```

  Java prototype

  ```
  public int ExampleMethod(String param1, int param2);
  ```

  In the above scenario parameter 2 ("5") will be cast to integer representation.

  Example of usage:

  ```
  # Kamailio
  java_method_exec("ExampleMethod", "Ljava/lang/String;I", "$mb", "$ml");

  # Java
  public int ExampleMethod(String SipMessageBuffer, int SipMessageLenght)
  {
    ... do something with buffer;
    return 1;
  }
  ```

- **Example 1.10. Signature: "ZB"**

  Kamailio prototype

  ```
  java_method_exec("ExampleMethod", "ZB", "true", "0x05");
  ```

  Java prototype

  ```
  public int ExampleMethod(boolean param1, byte param2);
  ```

  In the above scenario parameter 1 ("true") will be cast to boolean representation.

  Example of usage:

  ```
  # Kamailio
  java_method_exec("ExampleMethod", "ZB", "true", "0x05");

  # Java
  public int ExampleMethod(boolean flagSet, byte bFlag);
  {
    if (flagSet)
    {
     ... do something with flags;
    }
    return 1;
  }
  ```

java_staticmethod_exec(method, method_signature, [param1[, param2[, ...]]])

Executes a java static method *method*. Parameter *method_signature* is required.

- **Example 1.11. Signature: "V"**

  Kamailio prototype

  ```
  java_staticmethod_exec("ExampleMethod", "V");
  ```

  Java prototype

  ```
  public static int ExampleMethod();
  ```

  Example of usage:

  ```
  # Kamailio
  java_staticmethod_exec("ExampleMethod", "V");

  # Java
  public static int ExampleMethod()
  {
    ... do something;
    return 1;
  }
  ```

- **Example 1.12. Signature: "Ljava/lang/String;I"**

  Kamailio prototype

  ```
  java_staticmethod_exec("ExampleMethod", "Ljava/lang/String;I", "Hello world",
  ```

  Java prototype

  ```
  public static int ExampleMethod(String param1, int param2);
  ```

  In the above scenario parameter 2 ("5") will be cast to integer representation.

  Example of usage:

  ```
  # Kamailio
  java_staticmethod_exec("ExampleMethod", "Ljava/lang/String;I", "$mb", "$ml");

  # Java
  public static int ExampleMethod(String SipMessageBuffer, int SipMessageLenght)
  {
    ... do something with buffer;
    return 1;
  }
  ```

- **Example 1.13. Signature: "ZB"**

  Kamailio prototype

  ```
  java_staticmethod_exec("ExampleMethod", "ZB", "true", "0x05");
  ```

  Java prototype

  ```
  public static int ExampleMethod(boolean param1, byte param2);
  ```

  In the above scenario parameter 1 ("true") will be cast to boolean representation.

  Example of usage:

```
# Kamailio
java_staticmethod_exec("ExampleMethod", "ZB", "true", "0x05");

# Java
public static int ExampleMethod(boolean flagSet, byte bFlag);
{
  if (flagSet)
  {
   ... do something with flags;
  }
  return 1;
}
```

java_s_method_exec(method, method_signature, [param1[, param2[, ...]]])

Executes a java class synchronized method *method*. Parameter *method_signature* is required.

For more info see *Synchronized Methods* [http://docs.oracle.com/javase/tutorial/essential/concurrency/syncmeth.html]

- **Example 1.14. Signature: "V"**

  Kamailio prototype

  ```
  java_s_method_exec("ExampleMethod", "V");
  ```

  Java prototype

  ```
  public synchronized int ExampleMethod();
  ```

  Example of usage:

  ```
  # Kamailio
  java_s_method_exec("ExampleMethod", "V");

  # Java
  public synchronized int ExampleMethod()
  {
    ... do something;
    return 1;
  }
  ```

- **Example 1.15. Signature: "Ljava/lang/String;I"**

  Kamailio prototype

  ```
  java_s_method_exec("ExampleMethod", "Ljava/lang/String;I", "Hello world", "5")
  ```

  Java prototype

  ```
  public synchronized int ExampleMethod(String param1, int param2);
  ```

  In the above scenario parameter 2 ("5") will be cast to integer representation.

  Example of usage:

```
# Kamailio
java_s_method_exec("ExampleMethod", "Ljava/lang/String;I", "$mb", "$ml");

# Java
public synchronized int ExampleMethod(String SipMessageBuffer, int SipMessageL
{
  ... do something with buffer;
  return 1;
}
```

- **Example 1.16. Signature: "ZB"**

  Kamailio prototype

  ```
  java_s_method_exec("ExampleMethod", "ZB", "true", "0x05");
  ```

  Java prototype

  ```
  public synchronized int ExampleMethod(boolean param1, byte param2);
  ```

  In the above scenario parameter 1 ("true") will be cast to boolean representation.

  Example of usage:

  ```
  # Kamailio
  java_s_method_exec("ExampleMethod", "ZB", "true", "0x05");

  # Java
  public synchronized int ExampleMethod(boolean flagSet, byte bFlag);
  {
    if (flagSet)
    {
     ... do something with flags;
    }
    return 1;
  }
  ```

java_s_staticmethod_exec(method, method_signature, [param1[, param2[, ...]]])

Executes a java synchronized static method *method*. Parameter *method_signature* is required.

For more info see *Synchronized Methods* [http://docs.oracle.com/javase/tutorial/essential/concurrency/syncmeth.html]

- **Example 1.17. Signature: "V"**

  Kamailio prototype

  ```
  java_s_staticmethod_exec("ExampleMethod", "V");
  ```

  Java prototype

  ```
  public static synchronized int ExampleMethod();
  ```

  Example of usage:

```
# Kamailio
java_s_staticmethod_exec("ExampleMethod", "V");

# Java
public static synchronized int ExampleMethod()
{
  ... do something;
  return 1;
}
```

- **Example 1.18. Signature: "Ljava/lang/String;I"**

  Kamailio prototype

  ```
  java_s_staticmethod_exec("ExampleMethod", "Ljava/lang/String;I", "Hello world"
  ```

  Java prototype

  ```
  public static synchronized int ExampleMethod(String param1, int param2);
  ```

  In the above scenario parameter 2 ("5") will be cast to integer representation.

  Example of usage:

  ```
  # Kamailio
  java_s_staticmethod_exec("ExampleMethod", "Ljava/lang/String;I", "$mb", "$ml")

  # Java
  public static synchronized int ExampleMethod(String SipMessageBuffer, int SipM
  {
    ... do something with buffer;
    return 1;
  }
  ```

- **Example 1.19. Signature: "ZB"**

  Kamailio prototype

  ```
  java_s_staticmethod_exec("ExampleMethod", "ZB", "true", "0x05");
  ```

  Java prototype

  ```
  public static synchronized int ExampleMethod(boolean param1, byte param2);
  ```

  In the above scenario parameter 1 ("true") will be cast to boolean representation.

  Example of usage:

  ```
  # Kamailio
  java_s_staticmethod_exec("ExampleMethod", "ZB", "true", "0x05");

  # Java
  public static synchronized int ExampleMethod(boolean flagSet, byte bFlag);
  {
    if (flagSet)
    {
  ```

```
     ... do something with flags;
    }
    return 1;
  }
```

Java Module API

Minimal program skeleton

Example 1.20. Minimal program skeleton

```
import org.siprouter.*;
import org.siprouter.NativeInterface.*;

public class Kamailio extends NativeMethods
{
  /* Here you should specify a full path to app_java.so */
  static
  {
    System.load("/opt/kamailio/lib/kamailio/modules/app_java.so");
  }

  /* Constructor. Do not remove !!! */
  public Kamailio()
  {
  }

  /*
  This method should be executed for each children process, immediately after f
  Required. Do not remove !!!
  */
  public int child_init(int rank)
  {
    return 1;
  }
}
```