



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Rules System Deployment Guide

Rule Execution

12/17/2025

Rule Execution

The Rules Engine accepts REST requests from clients through a configured port. Clients that want to execute a rule package will connect to this port and send an HTTP POST message to:

`http: //{server-address:port}/{server-id}/knowledgebase/{packageName}`

This port is configured in the GRE application. See [Installing the GRE Component](#) for more information about how to configure this port.

The `server-id` is a configured value for the server and is not examined for the request. The `packageName` corresponds to the already deployed rule package that is to be evaluated.

The body of the HTTP request contains a knowledgebase request in either XML or JSON format. If JSON is used, the Content-Type HTTP header must be set to `application/json`. A successful response will contain a knowledgebase response message that contains the results of the evaluation.

The following schema defines the body of both the knowledgebase request and knowledgebase response message bodies.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">

    <xs:element name="knowledgebase-request">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="globals" type="globals" minOccurs="0"
maxOccurs="1"/>
                <xs:element name="inFacts" type="inFacts" minOccurs="0"
maxOccurs="1"/>
                <xs:element name="inOutFacts" type="inOutFacts"
minOccurs="0" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="knowledgebase-response">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="globals" type="globals" minOccurs="0"
maxOccurs="1"/>
                <xs:element name="outFacts" type="outFacts" minOccurs="0"
maxOccurs="1"/>
                <xs:element name="inOutFacts" type="inOutFacts" minOccurs="0"
maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="globals">
        <xs:element name="named-fact" type="named-fact" maxOccurs="unbounded"/>
    </xs:complexType>
</xs:schema>
```

```

</xs:complexType>
<xs:complexType name="inFacts">
  <xs:group ref="factGroup" maxOccurs="unbounded"/>
</xs:complexType>
<xs:complexType name="inOutFacts">
  <xs:group ref="factGroup" maxOccurs="unbounded"/>
</xs:complexType>
<xs:complexType name="outFacts">
  <xs:group ref="factGroup" maxOccurs="unbounded"/>
</xs:complexType>

<xs:group name="factGroup">
  <xs:choice>
    <xs:element name="named-fact"/>
    <xs:element name="anon-fact"/>
  </xs:choice>
</xs:group>

<xs:complexType name="named-fact">
  <xs:sequence>
    <xs:element name="id" type="id"/>
    <xs:element name="fact" type="fact"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="id">
  <xs:annotation>
    <xs:documentation>The identifier for a named fact</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<xs:complexType name="anon-fact">
  <xs:sequence>
    <xs:element name="fact" type="fact"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="fact">
  <xs:annotation>
    <xs:documentation>Contained elements are named after the fields of
the class
referred to by the class attribute. Element values are the values of
the fields
</xs:documentation>
  </xs:annotation>
  <xs:attribute name="class" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="\c+(\.\c+)*"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:schema>

```

In addition to the action resulting from evaluation of the conditions in a rule that executes, the GRE returns to the application a list of the names of each rule executed, and in addition, for decision tables, the name of each row that was executed. For the REST interface, the GRE returns an array of rule names within an `executionResult` block. For example:

```
<executionResult>
  <rulesApplied>
    <string>Rule-110 Dept ABC</string>
    <string>Rule-118 Process XYZ</string>
  </rulesApplied>
</executionResult>
```

For the ESP interface, the GRE returns the following two parameters on the ESP response:

- NumberOfRulesApplied - A count of the number of rules or decision tables rows that evaluated to true.
- RulesApplied - The names of the rules or decision tables rows that evaluated to true. The names are separated by semicolons. The rules are listed in the order of execution. If a rule is executed multiple times it will appear in the list multiple times.