

# RSB Performance Test API - Tasks #2400

## Generator API for a single RST type

09/23/2015 01:55 PM - J. Wienke

<b>Status:</b>	Resolved	<b>Start date:</b>	10/09/2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	H. Oestreich	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	0.1		
<b>Description</b>			
- frequency - user-defined payload (size)			
rst.classification.ClassifiedRegion2D			
g = new Generator(); g.addPayload(new Payload() {sfsdfsdf}.setTiming(new GridTiming(1Hz, 100 Hz, 10Hz))); g.addPayload(new Payload() {sfsdfsdf}.setTiming(new GridTiming(1Hz, 100 Hz, 10Hz))); g.execute();			
<b>Subtasks:</b>			
Tasks # 2407: Create Test Program which uses the API			<b>Resolved</b>

### History

#### #1 - 09/23/2015 01:56 PM - J. Wienke

- Description updated

#### #2 - 10/02/2015 05:04 PM - H. Oestreich

- % Done changed from 0 to 30

First version has been implemented and uploaded to the repo.  
No evaluation and tests done yet. Will be done next week.

#### #3 - 10/09/2015 02:26 PM - H. Oestreich

The current implementation looks like this:

```
final Generator generator = new Generator();

//Generation of the RST Object
[...]

// Generation of the payload
final Payload payload = new Payload() {};
payload.setData(clRe2D); // clRe2D = RST Object from Type ClassifiedRegion2D
payload.setScope(new Scope(scope));
final TimingGrid tg = new TimingGrid();
```

```
tg.addEntryAtEnd(new FrequencyDuration(1000, 15));
tg.addEntryAtEnd(new FrequencyDuration(5000, 30));
payload.setTiming(tg);

// Adding the generated payload to the generator and executing it
generator.addPayload(payload);
generator.execute();
```

This does not exactly look like what Johannes proposed in the ticket, but it has been tested successfully.

I did take a look at the Google Protocol Buffers / RST Generated Java Files, but I was not able to adapt my implementation to that style yet.

#### **#4 - 10/23/2015 05:43 PM - H. Oestreich**

- Status changed from New to In Progress

The API now got as simple as:

```
final Generator generator = new Generator();
generator.addTestPhase(createFrequencyScalingPhase());
generator.execute();
```

And an example for the creation of a test Phase:

```
final SimplePhase rpcCapture = new SimplePhase();
rpcCapture.setTiming(new OneTimeExecutionTiming());
rpcCapture.setTransaction(new RPCTransaction("capture",
    new Event(Long.class, new Long(0)),
    Factory.getInstance().createRemoteServer(scope)));
generator.addTestPhase(rpcCapture);
```

Waiting for the release of version 0.1 before closing this ticket.

#### **#5 - 12/11/2015 02:49 PM - H. Oestreich**

- Status changed from In Progress to Resolved