

Custom Script Jobs

Custom Script jobs replace the groovy script that is implemented in a system job to automate routine actions. There are a total of five Custom Script system jobs that are available on the Administration > Scheduled Jobs menu, named sequentially from Custom Script Job 1 to Custom Script Job 5. One file can be assigned to each custom job, but there can be multiple groovy scripts within each file. There are five custom jobs so that you can batch your groovy scripts to run at different intervals.

To use a Custom Script job

1. Create a groovy script file with the required logic.
2. Name the groovy script file with the same name as the Custom Script job.
3. Copy the groovy script file to the `%AGILIANCE_HOME%\Config\scripts` directory.
4. Restart the RiskVision Tomcat service or reload the server configuration.
5. Go to the **Administration > Scheduled Jobs**.
6. Check the box next to the Custom Script system job and click Activate. For example, if you have named the groovy script file as "CustomScriptJob1," then activate the Custom Script Job 1 system job. Note that there can be no spaces when naming the groovy script file.
7. Optional. Click the Custom Script system job to open its details page. And, click Reschedule to revise the execution time and to set the recurrence.

Example groovy script:

```
/**
```

This Groovy file should be placed in {customizations.folder}/scripts folder.

```
*/
```

```
import com.agiliance.admin.scheduler.CustomScriptJob.ScriptJob
```

```
import com.agiliance.web.customization.scripting.CustomizationBase
```

```
import java.lang.StringBuilder;
```

```
import java.util.Map;
```

```
/**
```

```
* This is an example of script class written in Groovy that can be used as a template for  
a script based job.
```

```
* The class must implement ScriptJob interface. If it extends CustomizationBase class,  
then the logging (info, warn, error) and other methods are also available.
```

```
*/
```

```
public class CustomScriptJob1 extends CustomizationBase implements ScriptJob {
```

```

/**
 * The method gets called from 'CustomScriptJob' system job.
 *
 * @param namedParams Job data
 *
 * @return
 */
public void execute(Map namedParams)
{
    // The following helper methods can be used for logging:
    info("CustomScriptJob: info() method!")

    // warn("CustomScriptJob: warn() method!")

    // error("CustomScriptJob: error() method!")

    // Log all job named parameters
    logAllNamedParameters(namedParams);
    // For more detailed info about all available helper methods take a look at CustomizationBase
    //Java class that this class extends.
}
private void logAllNamedParameters(Map namedParams) {
    StringBuilder sb = new StringBuilder();
    // Log all script job parameters
    sb.append("Job Named Parameters: [ ");
    for (Map.Entry e : namedParams.entrySet())
    {
        sb.append(e.getKey()).append(": ").append(e.getValue()).append(" ");
    }
    sb.append("]");
    info(sb.toString());
}
}

```