RunaWFE. Guide for Administrator scripts development.

Version 3.0

Intorduction

Scripts are located in a separate adminkit folder.

Under Windows, a script is launched by the following command: script-runner.bat

Under Linux, a script is launched by the following command: script-runner.sh

Script configuration is defined in a separate file (by default scripts/deploy-samples-script.xml)

Configuring script-runner.bat (script-runner.sh)

1. Assign the name of the user launching the script to the LOGIN variable (set LOGIN="Name")
2. Assign this user's password value to the PASSWORD variable (set PASSWORD="password")
3. The name of the file containing the script configuration can also be specified (script.xml by default). To do this, assign the file name value to the SCRIPT variable.

The script uses the wfescript_delegate.properties configuration file, located in the conf folder. This file specifies how the script connects to the workflow system. Updatable parameters:

- ru.runa.af.delegate.remote.provider.url – defines URL RMI/IIOP connections to the workflow server. Must have the jnp protocol and contain an address and a port number for the connection.

Transactions

There are 2 ways to configure transactions on script operations:

- using attribute **defaultTransactionScope** of the root element **workflowScript**

Valid attribute values are [**all** | **job**]. Value **all** (default behaviour) means that all script operations will be executed in single transaction. Value **job** means that each script operation will be executed in new transaction.

*Note. This attribute will be ignored in case of existance **transactionScope** in script.*

- using element **transactionScope** there are possible to define explicit bounds of transaction

Example (there are 2 transactions; deletion of the process instance 69 will not occur if archiving fails)

```
<?xml version="1.0" encoding="UTF-8" ?>
<workflowScript xmlns="http://runa.ru/xml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://runa.ru/xml workflowScript.xsd">
 <transactionScope>
  <archiveOldProcessInstances id="69" />
  <removeOldProcessInstances id="69" />
 </transactionScope>
 <transactionScope>
  <removeOldProcessInstances id="100" idTill="110"/>
 </transactionScope>
</workflowScript>
```

Note. In old versions single transaction per script was used.

Script Language

A script is a properly formatted XML-document with an upper-level tag <workflowScript>, where the body of the script is contained.

A script must contain an XML Schema declaration in the workflowScript.xsd file. See a sample of a schema declaration in the attached scripts/deploy-samples-script.xml. sample file.

Utility Elements:

Element	Description
namedIdentitySet	Creates a named set of elements
identity	identifies an element
executor	identifies executor
permission	enumerates permissions
botConfiguration	describes bot configuration
batchPresentation	describes BatchPresentation

Executors elements:

Element	Description
createActor	Creates an actor.
createGroup	Creates a group of actors.
deleteExecutor	Deletes an executor.
deleteExecutors	Deletes executors.
addExecutorsToGroup	Adds an executor into a group.
removeExecutorsFromGroup	Removes an executor from a group.

Process Definition:

Element	Description
deployProcessDefinition	Deploys the process definition

Permissions:

Element	Description
addPermissionsOnActor	Adds permissions on an actor.
setPermissionsOnActor	Sets permissions on an actor.
removePermissionsOnActor	Removes permissions on an actor.
addPermissionsOnGroup	Adds permissions on a group.
setPermissionsOnGroup	Sets permissions on a group.
removePermissionsOnGroup	Removes permissions on a group.
addPermissionsOnSystem	Adds permissions on a system.
setPermissionsOnSystem	Sets permissions on a system.
removePermissionsOnSystem	Removes permissions on a system.
addPermissionsOnDefinition	Adds permissions on a business process definition.
setPermissionsOnDefinition	Sets permissions on a business process definition.
removePermissionsOnDefinition	Removes permissions on a business process definition.
addPermissionsOnProcessInstances	Adds permissions on all existing business process instances.
setPermissionsOnProcessInstances	Sets permissions on all existing business process instances.
removePermissionsOnProcessInstances	Removes permissions on all existing business process instances.
removeAllPermissionsFromExecutor	Removes all permissions on executor

Botstations and Bots:

Element	Description
createBotStation	Creates new bot station
updateBotStation	Updates a bot station
deleteBotStation	Deletes a bot station
createBot	Creates new bot
updateBot	Updates a bot
deleteBot	Deletes a bot
addPermissionsOnBotStations	Adds permissions on bot stations
setPermissionsOnBotStations	Sets permissions on bot stations
removePermissionsOnBotStations	Removes permissions on bot stations
addConfigurationsToBot	Adds configurations to bot
removeConfigurationsFromBot	Removes configurations from bot
removeAllConfigurationsFromBot	Removes all configurations from bot

Administrative actions:

Element	Description
replicateBatchPresentation	Copies the batch presentation settings from one executor to all other executors
changeSubstitutions	Changes the substitution rules
removeOldProcessInstances	Removes old process instances from the system data base
removeOldProcessDefinitionVersion	Removes unused process definitions from the system data base
archiveOldProcessInstances	Archives old process instances
archiveOldProcessDefinitionVersion	Archives unused process definitions
retrieveOldProcessInstances	Retrieves process instances from archive
removeOldProcessDefinitionVersion	Retrieves process definitions from archive

New commands addition and updates of existing commands might happen without immediate documentation update. For the latest version of commands syntaxes see file workflowScript.xsd.

Detailed Description of Elements

executor

Attributes:

Attribute	*Description*	*Mandatory*
name	Actor name	Mandatory

Defines an executor element with a given name

permission

Attributes:

Attribute	*Description*	*Mandatory*
name	Permission name	Mandatory

This element specifies the type of permission on an object granted to a subject (see "RUNA WFE Administrator’s Guide" for more information) and contains only one attribute – Name. The valid values for permission are as follows:

Valid Values for permission Element:

Value	*Description*
permission.read	Permission to read.
permission.update_permissions	Permission to update permissions.
permission.login_to_system	Permission to login.
permission.change_self_password	Permission to change one's own password
permission.create_executor	Permission to create a user.
permission.deploy_definition	Permission to load a process definition into the system.
permission.update_executor	Permission to update properties (of an executor).
permission.update_actor_status	Permission to update executor status
permission.list_group	Permission to view (group members).
permission.add_to_group	Permission to add members into a group.
permission.remove_from_group	Permission to remove members from a group.
permission.redeploy_definition	Permission to modify (a business process definition in the system)
permission.undeploy_definition	Permission to unload (a business process definition from the system).
permission.start_process	Permission to start a business process.
permission.read_instance	Permission to read an instance (of a business process).
permission.cancel_instance	Permission to cancel an instance (of a business process).
permission.bot_station_configure	Permission to configure botstations.

botConfiguration

Attributes:

Name	*Description*	*Mandatory*
name	The name of the task for bot	Mandatory
handler	Class that can handle this task	Mandatory when adding a configuration (not required when deleting a configuration)
configuration	A path to the configuration file	Mandatory when adding a configuration (not required when deleting a configuration)

Identity

Attributes:

Name	*Description*	*Mandatory*
name	Element name	Mandatory

Defines executor or process definition with a given name

namedIdentitySet

Attributes:

Name	*Description*	*Mandatory*
name	Identity set name	Mandatory
type	The type of the set elements	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
identity	Adds and element	0 or more
namedIdentitySet	Adds a set of elements	0 or more

Creates a named set of the elements (executors or process definitions). All the set elements must be of the same type. This set is referenced by name in subsequent usage.

createActor

Attributes:

Attribute	*Description*	*Mandatory*
name	Actor name	Mandatory
fullName	Full name	Optional
description	Description	Optional
password	Actor password	Optional

Tag example:

```
<createActor name="ivanov" fullName="P. N. Ivanov" description="engineer"/>
```

createGroup

Attributes:

Attribute	*Description*	*Mandatory*
name	Group name	Mandatory
description	Description	Optional

Tag example:

```
<createGroup name="Accountants" description=”Financial staff”/>
```

deleteExecutor

Attributes:

Attribute	*Description*	*Mandatory*
name	Name executor to be deleted	Mandatory

Tag example:

```
<deleteExecutor name="ivanova" />
```

deleteExecutors

Embedded elements:

Element	*Description*	*Mandatory*
deleteExecutor	An executor to delete	1 or more

This element is used to delete several executors in one query. The effect is the same as if after repeated usage of deleteExecutor for each executor separately. But deleteExecutors is a faster way to do it.

Usage example:

```
  <deleteExecutors>
    <deleteExecutor name=”removed1”/>
    <deleteExecutor name=”removed2”/>
  </deleteExecutors>
```

addExecutorsToGroup

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the group to which members are added	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
executor	An executor is an actor or a group of actors	0 or more
namedIdentitySet	Named identity set of executors	0 or more

Tag example:

```
<addExecutorsToGroup name="HR staff">
  <executor name="ivanova"/>
  <executor name="petrova"/>
  <namedIdentitySet name="Inspectors"/>
</addExecutorsToGroup>
```

removeExecutorsFromGroup

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the group, from which members are removed.	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
executor	Executor (an actor or a group of actors)	0 or more
namedIdentitySet	Named identity set of executors	0 or more

Tag example:

```
<removeExecutorsFromGroup name="RS Inspectors">
  <executor name="ivanova"/>
  <executor name="petrova"/>
  <namedIdentitySet name="Inspectors"/>
</removeExecutorsFromGroup>
```

deployProcessDefinition

Attributes:

Attribute	*Description*	*Mandatory*
file	Path to the .par file with process definition.	Mandatory
type	Process type (default type is Script).	Optional

tag example:

```
<deployProcessDefinition file="c:\processes\Vacation.par" type=”vacations”/>
```

redeployProcessDefinition

Attributes:

Attribute	*Description*	*Mandatory*
definitionId	Process definition identifier	Mandatory
file	Path to the .par file with process definition.	Mandatory
type	Process type (default type is Script).	Optional

addPermissionsOnActor

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the actor, permissions are added on	Mandatory
executor	Name of executor, permissions are added to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<addPermissionsOnActor name="ivanov" executor="agents">
  <permission name="permission.read"/>
  <permission name="permission.update_permissions"/>
</addPermissionsOnActor>
```

setPermissionsOnActor

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the actor, permissions are set on	Mandatory
executor	Executor, permissions on the actor are granted to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<setPermissionsOnActor name="ivanov" executor="agents">
   <permission name="permission.read"/>
   <permission name="permission.update_permissions"/>
</setPermissionsOnActor >
```

removePermissionsOnActor

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of actor, on whom permissions are removed	Mandatory
executor	Executor, whose permissions on the actor are removed	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<removePermissionsOnActor name="ivanov" executor="agents">
   <permission name="permission.read"/>
</removePermissionsOnActor>
```

addPermissionsOnGroup

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the group, on which permissions are added on	Optional
executor	Name of executor, permissions are added to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	0 or more
namedIdentitySet	Name of the executors set to which the permissions added	0 or more

The following permission names are valid for groups:

- permission.read
- permission.update_permissions
- permission.update_executor
- permission.list_group
- permission.list_group
- permission.remove_from_group

The permissions are added to the executor defined in the mandatory executor attribute. This executor will get named permissions over the executors from the embedded element namedIdentitySet (if it is set) and the executor defined by name attribute (if it is set).

Tag example:

```
<addPermissionsOnGroup name="ASU" executor="ivanov">
  <permission name="permission.read"/>
  <permission name="permission.list_group"/>
  <permission name="permission.add_to_group"/>
  <permission name="permission.remove_from_group"/>
</addPermissionsOnGroup>
```

setPermissionsOnGroup

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the group, on which permissions are set	Optional
executor	Name of executor, permissions are granted to.	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	0 or more
namedIdentitySet	Name of the executors set to which the permissions added	0 or more

Tag example:

```
<setPermissionsOnGroup name="ASU" executor="ivanov">
  <permission name="permission.read"/>
  <permission name="permission.list_group"/>
  <permission name="permission.remove_from_group"/>
</setPermissionsOnGroup>
```

removePermissionsOnGroup

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the group, on which permissions are removed	Optional
executor	Executor, whose permissions are removed	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	0 or more
namedIdentitySet	Name of the executors set to which the permissions added	0 or more

Tag example:

```
<removePermissionsOnGroup name="Information Systems" executor="ivanov">
  <permission name="permission.list_group"/>
</removePermissionsOnGroup>
```

addPermissionsOnSystem

Attributes:

Attribute	*Description*	*Mandatory*
executor	Executor, permissions on the system are added to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

The following permission names are valid for system:

- permission.read
- permission.update_permissions
- permission.login_to_system
- permission.change_self_password
- permission.create_executor
- permission.deploy_definition
- permission.change_self_password

Tag example:

```
<addPermissionsOnSystem executor="All employees">
  <permission name="permission.read"/>
  <permission name="permission.login_to_system"/>
</addPermissionsOnSystem>
```

setPermissionsOnSystem

Attributes:

Attribute	*Description*	*Mandatory*
executor	Executor, permissions on the system are granted to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<setPermissionsOnSystem executor="All employees">
  <permission name="permission.read"/>
  <permission name="permission.login_to_system"/>
</setPermissionsOnSystem>
```

removePermissionsOnSystem

Attributes:

Attribute	*Description*	*Mandatory*
executor	Executor, whose permissions on the system are removed	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<removePermissionsOnSystem executor="All employees">
  <permission name="permission.login_to_system"/>
</removePermissionsOnSystem>
```

addPermissionsOnDefinition

Attributes:

Attribute	*Description*	*Mandatory*
name	Business process, permissions are added on	Mandatory
executor	Name ofxecutor, permissions are added to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

The following permission names are valid for definition:

- permission.read
- permission.update_permissions
- permission.undeploy_definition
- permission.redeploy_definition
- permission.start_process

- permission.read_instance
- permission.cancel_instance

Tag example:

```
<addPermissionsOnDefinition name="leave" executor="ivanov">
   <permission name="permission.read"/>
   <permission name="permission.start_process"/>
   <permission name="permission.read_instance"/>
</addPermissionsOnDefinition>
```

setPermissionsOnDefinition

Attributes:

Attribute	*Description*	*Mandatory*
name	Business process, permissions are set on	Mandatory
executor	Executor, permissions are granted to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<setPermissionsOnDefinition name="leave" executor="ivanov">
   <permission name="permission.read"/>
   <permission name="permission.update_permissions"/>
   <permission name="permission.redeploy_definition"/>
   <permission name="permission.undeploy_definition"/>
   <permission name="permission.start_process"/>
   <permission name="permission.read_instance"/>
</setPermissionsOnDefinition>
```

removePermissionsOnDefinition

Attributes:

Attribute	*Description*	*Mandatory*
name	Business process, on which permissions are removed	Mandatory
executor	Executor, whose permissions are removed	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<removePermissionsOnDefinition name="leave" executor="ivanov">
   <permission name="permission.undeploy_definition"/>
</removePermissionsOnDefinition>
```

addPermissionsOnProcessInstances

Attributes:

Attribute	*Description*	*Mandatory*
name	Business process, on whose instances permissions are added	Mandatory
executor	Name of executor, permissions are added to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

The following permission names are valid for process instances:

- permission.read
- permission.update_permissions
- permission.cancel_instance

Tag example:

```
 <addPermissionsOnProcessInstances name="leave" executor="HR staff">
   <permission name="permission.update_permissions" />
   <permission name="permission.cancel_instance" />
 </addPermissionsOnProcessInstances>
```

setPermissionsOnProcessInstances

Attributes:

Attribute	*Description*	*Mandatory*
name	Business process, on whose instances permissions are granted	Mandatory
executor	Executor, permissions are granted to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<setPermissionsOnProcessInstances name="leave" executor="HR staff">
   <permission name="permission.read" />
   <permission name="permission.update_permissions" />
   <permission name="permission.cancel_instance" />
</setPermissionsOnProcessInstances>
```

removePermissionsOnProcessInstances

Attributes:

Attribute	*Description*	*Mandatory*
name	Business process, on whose instances permissions are granted	Mandatory
executor	The executor, permissions are granted to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<removePermissionsOnProcessInstances name="leave" executor="HR staff">
   <permission name="permission.cancel_instance" />
</removePermissionsOnProcessInstances >
```

removeAllPermissionsFromExecutor

Attributes:

Attribute	*Description*	*Mandatory*
name	The name of the executor, permissions on whom must be revoked from all executors	Mandatory

Tag example:

```
<removeAllPermissionsFromExecutor name="ivanova"/>
```

removeAllPermissionsFromProcessDefinition

Attributes:

Attribute	*Description*	*Mandatory*
name	The name of the definition, permissions on which must be revoked from all executors	Mandatory

Tag example:

```
<removeAllPermissionsFromProcessDefinition name="Vacation"/>
```

removeAllPermissionsFromProcessInstances

Attributes:

Attribute	*Description*	*Mandatory*
name	The name of the definition of process instances, permissions on which must be revoked from all executors	Mandatory

Tag example:

```
<removeAllPermissionsFromProcessInstances name="Vacation"/>
```

createBotStation

Attributes:

Attribute	*Description*	*Mandatory*
name	Bot station name	Mandatory
address	Bot station address	Optional

Tag example:

```
<createBotstation name="botstation" address="localhost"/>
```

updateBotStation

Attributes:

Attribute	*Description*	*Mandatory*
name	Bot station name	Mandatory
newName	New bot station name	Optional
address	Bot station address	Optional

Tag example:

```
<updateBotstation name="botstation" newName="localhostBotStation" address="localhost"/>
```

deleteBotStation

Attributes:

Attribute	*Description*	*Mandatory*
name	Bot station name	Mandatory

Tag example:

```
<deleteBotstation name="botstation"/>
```

createBot

Attributes:

Attribute	*Description*	*Mandatory*
name	Bot name	Mandatory
botStation	Bot station name	Mandatory
startTimeout	Start timeout	Optional
password	Bot password	Optional

Tag example:

```
<createBot name="UniversalBot" botStation="botstation" startTimeout="200" password="123"/>
```

updateBot

Attributes:

Attribute	*Description*	*Mandatory*
name	Bot name	Mandatory
newName	New bot name	Optional
botStation	Bot station name	Optional
newBotStation	New bot station name	Optional
startTimeout	Start timeout	Optional
password	Bot password	Optional

Tag example:

```
<updateBot name="UniversalBot" startTimeout="0" password="321"/>
```

deleteBot

Attributes:

Attribute	*Description*	*Mandatory*
name	Bot name	Mandatory

Tag example:

```
<deleteBot name="UniversalBot"/>
```

addConfigurationsToBot

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the bot to add configuration	Mandatory
botStation	Botstation name	Optional

Embedded elements:

Element	*Description*	*Mandatory*
botConfiguration	Configuration to add	At least 1

Tag example:

```
<addConfigurationsToBot name="UniversalBot">
   <botConfiguration name="" handler="" configuration=""/>
</addExecutorsToGroup>
```

removeConfigurationsFromBot

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the bot to remove configuration	Mandatory
botStation	Botstation name	Optional

Embedded elements:

Element	*Description*	*Mandatory*
botConfiguration	Configuration to remove	At least 1

Tag example:

```
<removeConfigurationsFromBot name="UniversalBot">
   <botConfiguration name="doNothing"/>
</removeExecutorsFromGroup>
```

removeAllConfigurationsFromBot

Attributes:

Attribute	*Description*	*Mandatory*
name	Name of the bot to add configuration	Mandatory

Tag example:

```
<removeAllConfigurationsFromBot name="UniversalBot"/>
```

addPermissionsOnBotStations

Attributes:

Attribute	*Description*	*Mandatory*
executor	Executor, permissions on the system are added to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

The following permissions are allowed to be used for BotStation element:

- permission.read
- permission.update_permissions
- permission.bot_station_configure

Tag example:

```
<addPermissionsOnBotStations executor="All employees">
  <permission name="permission.read"/>
  <permission name="permission.configure_permissions"/>
</addPermissionsOnBotStations>
```

setPermissionsOnBotStations

Attributes:

Attribute	*Description*	*Mandatory*
executor	Executor, permissions on the system are granted to	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<setPermissionsOnBotStations executor="All employees">
   <permission name="permission.read"/>
</setPermissionsOnBotStations>
```

removePermissionsOnBotStations

Attributes:

Attribute	*Description*	*Mandatory*
executor	Executor, whose permissions on the system are removed	Mandatory

Embedded elements:

Element	*Description*	*Mandatory*
permission	Permission type	Optional

Tag example:

```
<removePermissionsOnBotStations executor="All employees">
   <permission name="permission.read"/>
 </removePermissionsOnBotStations>
```

replicateBatchPresentation

Attributes:

Attribute	*Description*	*Mandatory*
batchName	The name of the BatchPresentation copy	Mandatory
setActive	Set status (active or not) to the created BatchPresentation copy. Possible values: 'all' - set active for all profiles; 'changed' - active only if BatchPresentation is changed in profile; 'none' - not active for all profiles	Default value is 'none'
useTemplates	Defines if BatchPresentation templates should be used. If 'no', then if copied BatchPresentation is present, the substitution always occurs	Optional, default value 'yes'

Embedded Elements:

Attribute	*Description*	*Mandatory*
batchPresentation	BatchPresentation description	1 or more

The replicateBatchPresentation element must have only one child element batchPresentation with attribute nama='source' and any number of batchPresentation child elements with attirbute name='template'. The BatchPresentation replication occurs in the following cases:

1. The copied BatchPresentation doesn't exist yet.
2. useTemplates='no'.
3. useTemplates='yes' and the substituted BatchPresentation equals to one of the BatchPresentation templates.

Tag example:

```
<replicateBatchPresentation batchName="replicatedNew" setActive="all" useTemplates="yes">
  <batchPresentation name="source" actorName="attila" batchId="listTasksForm" batchName="replicateMe"/>
  <batchPresentation name="template" batchId="listTasksForm"/>
</replicateBatchPresentation>
```

batchPresentation

Attributes:

Attribute	*Description*	*Mandatory*
`name`	Element name	Mandatory
`actorName`	Actor name to whom batchPresentation belongs.	Optional
`batchId`	BatchPresentation type. (See allowed values in AFProfileStrategy and WFProfileStrategy java classes sources)	Mandatory
`batchName`	BatchPresentation name	Optional

If actorName and BatchName are not specified the default BatchPresentation for a given batchId is used.

changeSubstitutions

Embedded Elements:

Attribute	*Description*	*Mandatory*
executor	Executors whose substitution rules need to be changed	1 or more
delete	Substitution rules to be deleted	0 or more
add	Substitution rules to be added	0 or more

The must be at least one executor element present (more than on is also allowed). Executor can be an actor or a group.

Changes in substitutions will be applied to all the executors and all group members set via executor element.

element 'delete' attributes:

- orgFunc - optional
- criteria - optional (default value 'always')

element 'add' attributes:

- orgFunc - optional
- criteria - optional (default value 'always')
- isFirst - optional (default value 'true')
- isEnable - optional (default value 'true')

If orgFunc (and/or) criteria are set in 'delete' element then all substitution rules that match all set parameters are deleted. But if no parameters set all the substitution rules are deleted.

Substitution rules will be applied in the order in which they are added. The first added rule with isFirst=true and isEnable=true will be the first, the second will be second and etc. The same logic goes for rules with isFirst=false but only from the tail of the queue. The first added will be the last rule, the second - the one before last, etc. If orgfunction is not specified in the 'add' element, then terminator will be added.

Allowed replacements: %self_code% %self_id% %self_name%

Tag example:

```
<changeSubstitutions>
  <executor name="***"/>
  <delete orgFunc="***" criteria="***"/>
  <add orgFunc="***" criteria="***" isFirst="***" isEnabled="***"/>
</changeSubstitutions>
```

removeOldProcessInstances

Attribute	*Description*	*Mandatory*
onlyFinished	Remove only finished processes	No
endDate	Date of the process instance end must not be later then *endDate*	No
startDate	Date of the process instance start must not be later then *startDate*	No
name	Name of the process definition	No
version	Maximum process definition version number for which to remove instances	No
id	ID of the process instance to remove	No
idTill	If this parameter is set, then the process instances with ids in range between *id* parameter and *idTill* parameter will be removed	No
dateInterval	Remove the process instance start date of which is in the range between *startDate* and *endDate*	No

A process instance is removed only if all conditions set by parameters are fulfilled. If you don't set parameter 6 (*id*), then you must set parameter 2 or 3 (*endDate* or *startDate*) (or both)

Parameters:

1. *onlyFinished* (true/false) - If set to true (default value) then only finished process instances are to be deleted.

2. process finish date < *endDate*. If *dateInterval* parameter set to true, then *endDate* paremeter is used to set the last date (exclusive) of the interval for removed process instances start date.

3. process start date < *startDate*. If *dateInterval* parameter set to true, then *startDate* paremeter is used to set the first date (exclusive) of the interval for removed process instances start date.

4. *name* - business process name.

5. *version* - business process definition version. To use this parameter, set *name* (parameter 4) of the business process as well. All the process instances with this name and version <= *version* will be removed.

6. *id* of the process instance to be removed.

7. *idTill* - if this parameter present then *id* (parameter 6) shoult also be set.The business process instances with ids in the range between *id* and *idTill* (inclusive) will be removed.

8. *dateInterval* (true/false) If you set this parameter to true (false is default) the 2 and 3 parameters should be set as well. *StartDate* and *endDate* parameters will define the interval of dates for the process instance start date for the process instances to be removed.

Note. To set the date parameter use this format **yyyy-MM-ddThh:mm:ss**

removeOldProcessDefinitionVersion

Attribute	*Description*	*Mandatory*
name	Name of the process definition	Yes
version	Maximum process definition version number which to remove	No

Process definitions with the name set in the *name* parameter and with version <= *version* will be deleted **if**:

- There's no process instances for this definition versions in the database
- This is not the latest version of the business process definition

If these conditions fail the definition will not be removed If *version* parameter is not set, all versions will be deleted but the latest.

archiveOldProcessInstancees

Attributes and parameters of the element are the same as for removeOldProcessInstances.

Additional settings:

In wf_logic.properties set paremeter archive.db.enabled to true in order to be able to work with archive data base.

In hibernate.archive.cfg you should indicate archive database data source that is configured in JBoss server:

```
... <property name="hibernate.connection.datasource">java:/mssql2_archive_ds</property> ...
```

(where java:/mssql2_archive_ds is the name of the data source defined in xml file in ${JBOSS_HOME\server\default\deploy}). hibernate.archive.cfg is similar to hibernate.cfg, they differ only in data source property and are placed in the same directory in jboss server ${JBOSS_HOME\server\default\conf. When you first start server with hibernate.archive.cfg present and configured, all the necessary WFE tables are created in the archive database. If you want to drop all WFE tables from archive and create them anew afterwards, you should manually drop WFE_Constants table from the archive database, then restart jboss (RunaWFE server).

You can view archived processes data via web interface. In order to do it you will need a separate RunaWFE server, where in hibernate.cfg the hibernate.connection.datasource property is set to archive database data source name. You can not add new elements via web interface if you use certain databases (e.g. MSSQL).

archiveOldProcessDefinitionVersion

Attributes and parameters are the same as for removeOldProcessDefinitionVersion. Additional settings are the same as for archiveOldProcessInstances element.

retrieveOldProcessInstances

This feature supposed to retrieve process instance from archive database and to place it back to the original database. Unfortunately it works only with simple databases (MySQL) and not tested there. It definitely can not work with MSSQL. Attributes and parameters are the same as for removeOldProcessDefinitionVersion. Additional settings are the same as for archiveOldProcessInstances element.

retrieveOldProcessDefinitionVersion

This feature supposed to retrieve process instance from archive database and place it back to original one. Unfortunately it works only with simple databases (MySQL) and not tested there. It definitely can not work with MSSQL. This feature works only with simple databases (MySQL) but not tested there. It doesn't work with MSSQL. Attributes and parameters are the same as for removeOldProcessDefinitionVersion. Additional settings are the same as for archiveOldProcessInstances element.

stopProcessInstance

Attribute	*Description*	*Mandatory*
id	Process instance id	Yes

example:

<stopProcessInstance id=”1”/>

References

[1] http://www.gnu.org/licenses/lgpl.html)

Article Sources and Contributors

RunaWFE. Guide for Administrator scripts development. *Source*: http://wf.runa.ru/doc/index.php?oldid=738 *Contributors*: Dofs, Natkinnat, WikiSysop