Server Training
Table of Contents

Product Architecture
Server Administration
Workflow Sharing
Metanode templates
Remote Execution
The Webportal
SOAP Webservice interface
Architecture & Administration
What is the KNIME Server?
How does it work?

- KNIME Webportal
- KNIME Server
- VAADIN 7
- Glassfish Application Server
- KNIME Executor (a headless Analytics Platform)
- Java 7
Installation Overview

• Install Java
• Install KNIME Analytics Platform
• Create shared workflow repository
• Install Glassfish
• Configure application server
  – JVM Options
  – KNIME Server settings
  – Create local database
  – Create email resource
• Define users (or configure LDAP)
• Deploy KNIME Server
• Deploy KNIME Webportal
Administration Overview

Typical responsibilities of the KNIME Server Administrator:

• Perform server side updates (~2x per year)
  – Intermediate bug fix releases provided if necessary
• Structure the workflow repository
• Manager user access via permissions
Client Access
New features are needed for the KNIME Analytics Platform to access the Server.

Install the KNIME.com Client-Side Extensions from the KNIME Server update site.
Defining a new ServerSpace

Server connections are shown as “mount points” in the KNIME Explorer. To add a new mount point simply:

1. Click on the clipboard button in the KNIME Explorer.
2. Click New...
3. Configure a ServerSpace with your details

---

Image 1: KNIME Explorer window with a new ServerSpace being added.
Image 2: Preferences settings for configuring a ServerSpace.
Image 3: Selecting a new resource for display in the KNIME Explorer.
Image 4: Configuring a new ServerSpace with details.

---

Copyright © 2014 KNIME.com AG
ServerSpace as a shared repository

The Server provides an area in the Explorer for sharing work with your colleagues. Use workflow groups to organize your workflows, metanodes, and data files.

- Workflow group
- Data files
- Workflows
- Metanodes

To move resources simply drag and drop or copy and paste
Permissions

Access rights
Please set the access permissions for the item "/AdvancedCourse/Counting Loop 1"

The user owning the item: Owner name: rosaria.silipo

Access rights of the owner
- [ ] Read
- [ ] Modify
- [ ] Execute (applies only to workflows)

User group permissions
- [ ] Owner
- [ ] Groups

Access rights other users
- [ ] Read
- [ ] Modify
- [ ] Execute (applies only to workflows)

Available in KNIME Server

Group Permissions
User Group Permission for the Server Item
Users in one of the groups listed on the left have the assigned rights

Defined groups:
- [ ] << Add
- [ ] Remove

Define (or change) the rights for a group:
- [ ] Group name
- [ ] Enter the name of a group (must be valid)
- [ ] marketing

Permissions for users in this group
- [ ] Read
- [ ] Modify
- [ ] Execute (applies only to workflows)

OK Cancel
Remote Execution and Scheduling

Available in KNIME Server
Workflow Jobs

Remotely executed workflows are run as “Jobs”

- A workflow job is a copy of the workflow with specific settings and data.
- Jobs tied to a particular version of a workflow
- Orphaned jobs are colored red
- Jobs have messages (e.g. successful, or failure)
- Can be saved as workflow for data provenance and debugging (right-click save as)
Demo
The KNIME Webportal
Webportal Overview

- Publish workflows to be run from a standard web browser
- KNIME application not required to execute the workflow.
- Enables “self service” model for analytics deployment
- Only a subset of the complexity exposed to the workflow consumer
Quickforms

- Quickforms translated to UI elements in Webportal
- As workflow executes, Webportal searches for „active“ quickforms and displays them to the user
- Results in a stepwise execution of the workflow with only decorated node settings being exposed.
Quickforms in a Metanode
Quickform nodes for Web UI creation
Run a Workflow via Web Browser

Step 1
Select workflow to run

Step 2
Run settings and first workflow parameters

Step 3, ...
More workflow parameters

Last Step
See results

Available in KNIME Server
Webportal links to workflows

http://<Server_Address>:<port>/<ContextRoot>/#/<ItemPath>&<WorkflowParameters>

<ItemPath> = The path to a workflow, workflow group, or workflow job.

(A workflow job is referenced with its ID like: WorkflowGroup/Workflow?exec=job_id)

<WorkflowParameters> can appear in any order, but have to be after the <ItemPath>. Parameters are always appended with a leading '&'

Available Workflow Parameters

&single - Single mode: hides the workflow tree, so that only the selected item is visible.

&run - Auto run: If appended to workflow path, a new job is automatically executed.

&wv:<name>=<value> - Set variable: Sets the named workflow variable to the specified value.

&cr:<name>=<user>:<password> - Set credentials: Sets the user name and password for the named workflow credentials. Please note that the credentials and user name cannot contain the “=” and “;” symbols and that the password is transmitted in plain text.

&emails=sample@mail.com – Enable email notification: enables email notification and sets the specified comma separated list of email addresses.

&formats=<formats> - Set report formats: sets the report formats included as attachments in the notification email specified by a comma separated list. Available formats are: pdf (enabled by default), html, doc, xls, ppt, ps, odt, ods and odp.

A complete URL might look like:

http://localhost:8080/webportal/#/demo/file%20to%20csv&wv:title=foo&emails=sample@mail.com&run&single
Demo
Web service access
KNIME Web service Interface

• SOAP Based, Installed with the KNIME Server application

• Supports authentication

• Execute workflows from within KNIME or other applications

• Code examples available upon request
## KNIME Web service Interface API

<table>
<thead>
<tr>
<th>Operation (Method)</th>
<th>Input</th>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>listWorkflows</td>
<td>&lt;none&gt;</td>
<td>String[] – list of available workflows</td>
<td>Get all workflow paths of workflows visible to the current user</td>
</tr>
<tr>
<td>listWorkflowCredentials</td>
<td>String workflow path</td>
<td>WorkflowCredentials[] – list of all workflow credentials defined on this workflow</td>
<td>Get a list of workflow credentials for a given workflow</td>
</tr>
<tr>
<td>listWebServiceVariables</td>
<td>String workflow path</td>
<td>WebServiceVariable[] – list of all workflow variables with their default assignment</td>
<td>Get a list of workflow variables for a given workflow</td>
</tr>
<tr>
<td>invokeSync</td>
<td>String workflow path</td>
<td>WorkflowStatus – status of flow after execution</td>
<td>Execute a workflow, wait for it to finish.</td>
</tr>
<tr>
<td></td>
<td>WebServiceVariable[]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>list of flow variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>invokeAsync</td>
<td>String workflow path</td>
<td>String job ID – status of flow after execution</td>
<td>Execute a workflow, get job ID representing pending task</td>
</tr>
<tr>
<td></td>
<td>WebServiceVariable[]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>list of flow variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>getStatus</td>
<td>String job ID</td>
<td>WorkflowStatus – status of workflow</td>
<td>Get status of a running job</td>
</tr>
<tr>
<td></td>
<td>WebServiceVariable[]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>list of flow variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>generateReport</td>
<td>String Job ID</td>
<td>byte[] – content of the generated report</td>
<td>Generates the report associated with the executed flow (if any)</td>
</tr>
<tr>
<td></td>
<td>String reportformat</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(&quot;pdf&quot;, &quot;ppt&quot;, &quot;doc&quot;, etc.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>discard</td>
<td>String job ID</td>
<td>&lt;none&gt; – Cancel and/or discard a running or completed job</td>
<td></td>
</tr>
</tbody>
</table>
Programmatic access

URL wsdlURL = ...;

QName name = new QName("http://www.knime.com/soap",
   "GenericWorkflowService");

Service knimeService = Service.create(wsdlURL, name);

GenericWorkflowWebService accessor =
   knimeService.getPort(GenericWorkflowWebService.class);

BindingProvider bin = ((BindingProvider) accessor);

Map<String, Object> requestContext = bin.getRequestContext();

requestContext.put(BindingProvider.USERNAMEPROPERTY, user);

requestContext.put(BindingProvider.PASSWORDPROPERTY, pass);

accessor.listWorkflows();
accessor.invokeAsync(...);
SOAP Nodes

- Use to access KNIME Web services (or others)
- Supports workflow credentials
Demo