

Software Engineering Project (2IP40)

Project Group 1

Software User Manual

version 0.1.0 (Internally Accepted), 14 June 2006



Project Team:	Sven Bego	0550191
	Roel Coset	0548132
	Robert Leeuwestein	0546746
	Maarten Leijten	0547649
	Ivo van der Linden	0547632
	Joery Mens	0547515
	Marcel Moreaux	0499480
	Tim Muller	0547961
Project Manager:	Tom Kleijkers	0515015
Senior Manager:	L. Somers	TU/e HG 7.83
Advisor:	Y.Usenko	TU/e HG 5.71
Customer:	C. Plevier	Dutch Space
	H. de Wolf	Dutch Space

Abstract

This document is the Software User Manual (SUM) for the SPINGRID project and was made according to the software engineering standard described in [ESA] provided by European Space Agency (ESA). The Software User Manual (SUM) instructs how to install and use the SPINGRID software. This project is part of the Software Engineering Project course (2IP40) at the Technische Universiteit Eindhoven (TU/e).

Contents

1	Introduction	6
1.1	Intended readership	6
1.2	Applicability	6
1.3	Purpose	6
1.4	How to use this document	6
1.5	Related documents	7
1.6	Conventions	7
1.7	Problem reporting	7
2	Overview	8
3	Tutorial	9
3.1	Installing the software	9
3.1.1	Building the software	9
3.1.2	Installation	10
3.2	Uninstalling the software	11
3.3	A quick start with the client	11
3.4	A quick start with the agent	12
3.5	A quick start with the dispatcher	12
4	Reference	13
4.1	Client operations	13
4.1.1	add app	13
4.1.2	add data	14
4.1.3	add job	15
4.1.4	add project	16

4.1.5	add user	16
4.1.6	allow private	17
4.1.7	assign	18
4.1.8	disallow private	19
4.1.9	distrust	20
4.1.10	info	21
4.1.11	list apps	22
4.1.12	list data	23
4.1.13	list jobs	25
4.1.14	list projects	26
4.1.15	list users	27
4.1.16	passwd	28
4.1.17	remove	28
4.1.18	trust	29
4.1.19	unassign	30
4.2	Agent operations	31
4.2.1	add interval	31
4.2.2	exit	32
4.2.3	help	32
4.2.4	list projects	33
4.2.5	remove interval	33
4.3	Dispatcher operations	34
4.3.1	exit	34
A	Error messages and recovery procedures	35
B	Glossary	36

Document Status Sheet

Document Title	Software Transfer Document
Document Identification	SPINGRID/Documents/product/SUM/0.1.0
Author(s)	S. Bego, R. Leeuwstein
Version	0.1.0
Document Status	draft / <u>internally accepted</u> / conditionally approved / approved

Version	Date	Author(s)	Summary
0.0.1	22-05-2006	R. Leeuwstein	Document creation
0.0.2	12-06-2006	S. Bego, R. Leeuwstein	Version for internal review
0.1.0	14-06-2006	R. Leeuwstein	Internal approved

Document Change Report

Document Title	Software Transfer Document
Document Identification	SPINGRID/Documents/product/SUM/0.1.0
Date of Changes	N/A

10

Chapter 1

Introduction

1.1 Intended readership

This document covers the use for the following users of the SPINGRID system:

- 15 • the system administrator
- the project administrators
- the job providers
- the application providers
- the data providers
- 20 • the resource providers

1.2 Applicability

This Software User Manual (SUM) applies to the SPINGRID software, version 0.1.

1.3 Purpose

The purpose of the SUM is to assist the user in installing and using the SPINGRID software.

25 1.4 How to use this document

- Chapter 2 gives an overview of the SPINGRID software.
- Chapter 3 contains tutorials for common tasks that enable users to get started quickly.
- Chapter 4 gives a reference of the complete SPINGRID software.

- Appendix A includes a list of all error messages and recovery procedures.
- 30 • Appendix B contains the glossary.

1.5 Related documents

[ESA]	<i>ESA Software Engineering Standards (ESA PSS-05-0 Issue 2)</i> , ESA Board for Software Standardization and Control (BSSC), 1991
-------	--

1.6 Conventions

None.

1.7 Problem reporting

- 35 Since the SPINGRID team will be dissolved after completion of the SPINGRID project, the issue of problem reporting is left to Dutch Space.

Chapter 2

Overview

The software implements a computational grid. This grid is able to execute jobs when it
40 receives an application accompanied by a set of data files. By hiding the complexity of grid
technology which makes the system easy to use.

Chapter 3

Tutorial

3.1 Installing the software

45 3.1.1 Building the software

Before the software applications can be used, they need to be build from the source. Below is explained how to perform this process for the dispatcher. For the client and agent, the process is analogous.

1. Install Eclipse The easiest way to build the dispatcher from source is using *Eclipse SDK*. Eclipse SDK is a development environment and is a open-source project. The latest
50 version of this software can be obtained at <http://www.eclipse.org/downloads/>. After the download is completed, install the software using the instructions provided by the installer. Note that version 3.1 is used by the SPINGRID team. A different version could give problems.

2. Download the mysql driver for Java The dispatcher needs to communicate with a
55 mysql database and depends on *MySQL Connector/J*. The source of this driver needs to be downloaded from <http://dev.mysql.com/downloads/connector/j/>. Note that version 3.1 is used by the SPINGRID team. A different version could give problems.

3. Import the Fat Jar Eclipse Plug-in To include MySQL Connector/J, Eclipse needs a plug-in. After Eclipse is started, click *Help, Software Updates* and finally *Find And Install*.
60 Select *Search for new features to install* and click *next*. Click *New Remote Site* and enter the following URL: <http://kurucz-grafika.de/fatjar>. Choose as name *Fatjar* and then click *ok* to add the URL. Make sure only *Fatjar* is selected and click *finish*. The Fat Jar plug-in is detected and the instructions on screen can be followed to install the plug-in.

4. Create a new project from the source In Eclipse, click *File, New* and then *Project*.
65 Choose *Java Project* from the list and click *Next*. In the project name field, enter *Dispatcher*,

select *Create new project from existing source* and choose the root directory where the source files of the dispatcher are located. Click *Finish* to complete the process.

5. Import the mysql library Select the newly created project in the package explorer, choose *Project* and then *Properties* from the menu. Choose *Java Build Path* on the left and
 70 select the tab *Libraries*. Click the button *Add External JARs* and select *mysql-connector-java-x.x.xx-bin.jar* in the archive you downloaded. Next click on the tab *Order and Export* and check the imported jar. Click *OK* to close the window.

6. Build the project Click with the right mouse button on the project in the *Package Explorer* and select *Build Fat Jar*. In the dialog box that appears, choose a jar-name (e.g.
 75 *Dispatcher*), select the main class (*Dispatcher.java*) and start the building process by clicking *Finish*.

The process is analogous for the client and agent, except for the steps including MySQL and Fat Jar which can be skipped. Instead, these applications can be build by choosing *Export*
 80 instead of *Build Fat Jar* from the dropdown menu in the package explorer. The instructions on the screen can be followed to complete the normal build process.

3.1.2 Installation

After the source has been build, the jar-file created can be copied to any location and the programs are almost ready to be used. Before the first start it is necessary to create a
 85 configuration file named *options* in the same directory as the jar-file. For the agent an additional *properties* file needs to be created. In the next subsections you can find what to put in these configuration files. Note that text between the symbols < and > is only comment of what needs to be put there. The symbols < and > should not used in the files.

Client

90 file options

proxy_address=<the proxy address when used>

password=<your password>

http_port=<the port of the dispatcher to which to connect to (this would normally be port
 80)>

95 dispatcher_address=<the address of the dispatcher>

username=<your username in the system>

Agent

file options

100 proxy_address=<the proxy address when used>

```

http_port=<the port of the dispatcher to which to connect to (this would normally be port
80)>
proxy_port=<the port that used by the proxy>
dispatcher_address=<the address of the dispatcher>
105 agent_id=<this must be a unique identifier in the SPINGRID system>

```

file properties

```

FileSystem_NORMAL_MountPoint=<working directory for the agent>
FileSystem_NORMAL_Capacity=<available space in the working directory, given in bytes>
110 FileSystem_TMP_MountPoint=<temporary directory for the agent>
FileSystem_TMP_Capacity=<available space in the temporary directory, given in bytes>
ExclusiveExecution=<this may be true or false but has no effect>
OperatingSystem_Name=<operating system on which the agent is running e.g. Windows_XP>
OperatingSystem_Version=<version of the operating system e.g. SP2>
115 CPUArchitecture=<architecture on which the agent is running e.g. x86>
CPUSpeed=<cpu speed in hertz>
CPUCount=<total amount of available cpus>
NetworkBandwidth=<total amount of available bandwidth, given in bits per second>
PhysicalMemory=<total amount of available physical memory, given in bytes>
120 VirtualMemory=<total amount of available virtual memory, given in bytes>

```

Dispatcher**file options**

```

database_name=<name of the user that has full privileges over the database>
database_pass=<password of the user>
125 database_address=<address to the database e.g. jdbc:mysql://pcwin509.softeng.tue.nl/spingrid>
http_port=<port on which HTTP runs>
max_connections=<maximum number of simultaneous connections allowed>

```

3.2 Uninstalling the software

All the applications can be easily uninstalled by removing the jar-files.

130 3.3 A quick start with the client

After installing, the client can be started with "java -jar sgclient.jar". A list of available commands should be returned. More info of a command can be found by typing "java -jar sgclient.jar help" followed by a command. For an example "java -jar sgclient.jar help info".

3.4 A quick start with the agent

135 After installing, the agent can be started with "java -jar sgent.jar". A list of available commands should be returned. More info of a command can be found by typing "help" followed by a command. For an example "help add interval".

3.5 A quick start with the dispatcher

140 After installing, the dispatcher can be started with "java -jar sgdispatcher.jar". A list of available commands should be returned.

Chapter 4

Reference

The client, agent and dispatcher operations are described in this chapter. The description of the syntax of the operations needs an explanation:

- 145 • [arg]: Parameters between the symbols "[" and "]" mean that these are optional. Note that the symbols itself are never used.
- (arg0 | arg1 | ... | argn): Parameters between the symbols "(" and ")" and the symbol " | " between those parameters, mean that one of the parameters from arg0 to argn must be chosen. Note that the symbols itself are never used.

150 4.1 Client operations

4.1.1 add app

Functional description This operation adds an application to the SPINGRID system.

Formal description

- 155 • Syntax: add app fil:<filename> as <applicationname> <description>
- Parameters:
 - <filename> (Required) : This parameter is the name of file that is added as an application.
 - <applicationname> (Required) : This parameter is the name of the application that is added. It must be unique. Spaces are not allowed.
 - 160 – <description> (Required) : This parameter is the description of the application that is added.

Examples

- add app fil:<testapp.xml> as testapp this is a test application

165 **Possible errors**

- You do not have the role application provider.
- An application with name <applicationname> already exists.

Related operations remove**4.1.2 add data**

170 **Functional description** This operation adds a dataset to the SPINGRID system.

Formal description

- Syntax: add data (url:<url> | fil:<filename>) as <datasetname> <description>

- Parameters:

- 175 – <url> (Required when chosen) : This parameter is the URL of a file that is added. Note that more URL's are allowed in the syntax (see the examples). An URL must point to a file.
- 180 – <filename> (Required when chosen) : This parameter is the name of file that contains URL's which point to the files that are added. An URL must point to a file.
- <datasetname> (Required) : This parameter is the name of the dataset that is added. It must be unique. Spaces are not allowed.
- <description> (Required) : This parameter is the description of the dataset that is added.

185 **Examples**

- add data fil:URLs.txt as Football This is a football dataset
- add data url:http://www.medicalresearch.com/data1.dat
url:http://www.medicalresearch.com/data2.dat as MedicalResearch This is a medical research dataset

190 **Possible errors**

- You do not have the role data provider.
- A dataset with name <dataname> already exists.

Related operations remove**4.1.3 add job**195 **Functional description** This operation adds a job to the SPINGRID system.**Formal description**

- Syntax: add job fil:<filename> to pro:<projectname>
- Parameters:
 - 200 – <filename> (Required) : This parameter is the name of the file that describes the job. This must be a JSDL-file.
 - <projectname> (Required) : This parameter is the name of the project where the job is added.

Examples

- 205
- add job fil:sumcalculation.jsdl to pro:MedicalResearch

Possible errors

- Project <projectname> does not exist.
- You do not have the role job provider in project <projectname>.
- A job with name <jobname> already exists.
- 210 • Application <applicationname> does not exist.
- Your project is not approved to use application <applicationname>.
- The owner of the application is no longer an application provider. His applications can not be used.
- The owner of the application is not trusted by a project admin in project <projectname>.
- 215 His applications can not be used.
- Your project is not approved to use dataset <dataset>.

- The owner of the data set is no longer a data provider. His data sets can not be used.
- The owner of the data set is not trusted by a project admin in project <projectname>. His data sets can not be used.

220 **Related operations** remove

4.1.4 add project

Functional description This operation adds a project to the SPINGRID system.

Formal description

- Syntax: add project <projectname> with usr:<username>

225

- Parameters:

- <projectname> (Required) : This parameter is the name of project that is added. Spaces are not allowed.
- <username> (Required) : This parameter is the name of the user that becomes project admin of the added project. The user must exist.

230

Examples

- add project MedicalResearch with usr:Henk

Possible errors

- You do not have the role system admin.
- A project with name <projectname> already exists.
- User <username> does not exist.

235

Related operations remove

4.1.5 add user

Functional description This operation adds an user to the SPINGRID system.

240 **Formal description**

- Syntax: `as rol:<rolename> add user <username> <password>`
- Parameters:
 - `<rolename>` (Required) : This parameter is the name of role that you have. This can either be "sysadmin" or "projadmin"
 - `<username>` (Required) : This parameter is the name of the user that is added. Spaces are not allowed.
 - `<password>` (Required) : This parameter is the password of the user that is added. Spaces are not allowed.

250 **Examples**

- `as rol:sysadmin add user Henk kneH`
- `as rol:projadmin add user Piet teiP`

Possible errors

- You do not have the role project admin.
- You do not have the role system admin.
- An user with username `<username>` already exists.

Related operations `remove`**4.1.6 allow private**

Functional description This operation allows an user to use private applications or datasets in a project.

Formal description

- Syntax: `allow private <appsordata> for usr:<username> in pro:<projectname>`
- Parameters:
 - `<appsordata>` (Required) : This parameter can be "apps" (to allow the user to use private applications) or "data" (to allow the user to use private datasets).
 - `<username>` (Required) : This parameter is the name of the user that is allowed to use applications or datasets in the project.
 - `<projectname>` (Required) : This parameter is the name of the project were the user is trusted in.

Examples

- allow private apps for usr:Henk in pro:MedicalResearch
- allow private data for usr:Henk in pro:MedicalResearch

Possible errors

- 275
- Project <projectname> does not exist.
 - You do not have the role project admin in project <projectname>.
 - User <username> does not exist.
 - User <username> does not have the role JobProvider in project <projectname>.
- 280
- You, as project admin of project <projectname>, already allowed <username> to use his own applications for jobs in that project.
 - You, as project admin of project <projectname>, already allowed <username> to use his own datasets for jobs in that project.

Related operations disallow private

4.1.7 assign

285 **Functional description** This operation assigns a role to an user.

Formal description

- Syntax: assign rol:<rolename> to usr:<username> [in pro:<projectname>]
- Parameters:
 - 290 – <rolename> (Required) : This parameter is the name of role that is assigned to the user.
 - <username> (Required) : This parameter is the name of the user where the role is assigned to.
 - 295 – <projectname> (Optional) : This parameter is the name of the project where the user as the role is assigned to. This parameter is only necessary when assigning the role project admin or job provider to an user.

Examples

- assign rol:appprov to usr:Henk
- assign rol:jobprov to usr:Piet in pro:MedicalResearch

300 **Possible errors**

- You do not have the role system admin.
- User <username> does not exist.
- Project <projectname> does not exist.
- You do not have the role of project admin in project <projectname>.
- 305 • User <username> is already an application provider.
- User <username> is already a data provider.
- User <username> is already a job provider in project <projectname>.
- User <username> is already a project admin of project <projectname>.

Related operations unassign310 **4.1.8 disallow private**

Functional description This operation disallows an user to use private applications or datasets in a project.

Formal description

- Syntax: disallow private <appsordata> for usr:<username> in pro:<projectname>
- 315 • Parameters:
 - <appsordata> (Required) : This parameter can be "apps" (to disallow the user to use private applications) or "data" (to disallow the user to use private datasets).
 - <username> (Required) : This parameter is the name of the user that is disallowed to use applications or datasets in the project.
 - 320 – <projectname> (Required) : This parameter is the name of the project were the user is not trusted in.

Examples

- disallow private apps for usr:Henk in pro:MedicalResearch
- 325 • disallow private data for usr:Henk in pro:MedicalResearch

Possible errors

- Project <projectname> does not exist.
- You do not have the role project admin in project <projectname>.
- User <username> does not exist.
- 330 • User <username> does not have the role JobProvider in project <projectname>.
- User <username> does not have the right to use his own applications for jobs in project <projectname>.
- User <username> does not have the right to use his own datasets for jobs in project <projectname>.

335 **Related operations** allow private

4.1.9 distrust

Functional description This operation lets an user distrust an application, dataset or user for a project.

Formal description

- 340 • Syntax: distrust (app:<applicationname> | <datasetname> | <username> as rol:<rolename>) [for <projectname>]
- Parameters:
 - <applicationname> (Required when chosen) : This parameter is the name of the application that is distrusted.
 - 345 – <datasetname> (Required when chosen) : This parameter is the name of the dataset that is distrusted.
 - <username> (Required when chosen) : This parameter is the name of the user that is distrusted.
 - 350 – <rolename> (Required when chosen) : This parameter is the name of the role of the user that is distrusted.
 - <projectname> (Required when chosen) : This parameter is the name of the project where the user is distrusted in. This parameter is required when distrusting an application or data provider.

355 Examples

- distrust app:testapp for pro:testproject
- distrust usr:Henk as rol:appprov

Possible errors

- User <username> does not exist.
- 360 • User <username> does not have the role job provider in project <projectname>.
- Project <projectname> does not exist.
- You do not have the role application provider.
- You do not have the role data provider.
- You do not have the role project admin in project <projectname>.
- 365 • DataSet <datasetname> does not exist.
- Application <applicationname> does not exist.
- DataSet <datasetname> is not yours.
- Application <applicationname> is not yours.
- You, as application provider did not allow project <projectname> to use your applica-
370 tion <applicationname>.
- You, as data provider did not allow project <projectname> to use your dataset <datasetname>.
- Project <projectname> does not exist.

Related operations trust**4.1.10 info**

375 **Functional description** This operation return information of an entity in the SPINGRID system.

Formal description

- Syntax: info (app:<applicationname> | dat:<datasetname> | job:<jobname> | pro:<projectname> |
380 | usr:<username>)
- Parameters:
 - <applicationname> (Required when chosen) : This parameter is the name of the application from which the info is returned
 - <datasetname> (Required when chosen) : This parameter is the name of the
385 dataset from which the info is returned
 - <jobname> (Required when chosen) : This parameter is the name of the job from which the info is returned

- <projectname> (Required when chosen) : This parameter is the name of the project from which the info is returned
- 390 – <username> (Required when chosen) : This parameter is the name of the user from which the info is returned

Examples

- info pro:MedicalResearch
- info usr:Henk

395 Possible errors

- User <username> does not exist.
- Project <projectname> does not exist.
- DataSet <datasetname> does not exist.
- Application <applicationname> does not exist.
- 400 • Job <jobname> does not exist.

Related operations None

4.1.11 list apps

Functional description This operation returns a list of applications. This operation can be executed either as a project admin, a system admin, a job provider or as an application
405 provider.

- Project admin: As a project admin, it returns a list of all applications that can be used in one or more of your projects. If a project is given it returns the applications that can be used in that specific project.
- 410 • System admin: As system admin, it returns a list of all applications in the system. If a project is given it returns the applications that can be used in that specific project.
- Job provider: As job provider, it returns a list of all applications that can be used in your jobs. If a project is given it returns all applications that can be used in your jobs in that specific project.
- 415 • Application provider: As application provider it returns a list of the applications you have provided. If a project is given it returns all applications that you have trusted in the specific project.

Formal description

- Syntax: `as rol:<rolename> list apps [in pro:<projectname>]`

- 420 • Parameters:

- `<rolename>` (Required) : This parameter is the name of the role you have. This can either be "projadmin", "sysadmin", "jobprov" or "approv".
- `<projectname>` (Optional) : This parameter is the name of the project. See the functional description of this operation for more information.

425 **Examples**

- `as rol:approv list apps`
- `as rol:sysadmin list apps in pro:MedicalResearch`

Possible errors

- You do not have the role project admin.
- 430 • You do not have the role system admin.
- You do not have the role job provider.
- You do not have the role application provider.
- Project `<projectname>` does not exist.
- You do not have the role project admin in project `<projectname>`.
- 435 • You do not have the role job provider in project `<projectname>`.

Related operations list data**4.1.12 list data**

Functional description This operation returns a list of datasets. This operation can be executed either as a project admin, a system admin, a job provider or as an data provider.

- 440 • Project admin: As a project admin, it returns a list of all datasets that can be used in one or more of your projects. If a project is given it returns the datasets that can be used in that specific project.
- System admin: As system admin, it returns a list of all datasets in the system. If a project is given it returns the datasets that can be used in that specific project.

- 445
- Job provider: As job provider, it returns a list of all datasets that can be used in your jobs. If a project is given it returns all datasets that can be used in your jobs in that specific project.
 - Data provider: As data provider it returns a list of the datasets you have provided. If a project is given it returns all applications that you have trusted in the specific project.

450 **Formal description**

- Syntax: `as rol:<rolename> list data [in pro:<projectname>]`
 - Parameters:
 - `<rolename>` (Required) : This parameter is the name of the role you have. This can either be "projadmin", "sysadmin", "jobprov" or "datprov".
 - `<projectname>` (Optional) : This parameter is the name of the project. See the functional description of this operation for more information.
- 455

Examples

- `as rol:datprov list data`
 - `as rol:sysadmin list data in pro:MedicalResearch`
- 460

Possible errors

- You do not have the role project admin.
 - You do not have the role system admin.
 - You do not have the role job provider.
 - You do not have the role data provider.
 - Project `<projectname>` does not exist.
 - You do not have the role project admin in project `<projectname>`.
 - You do not have the role job provider in project `<projectname>`.
- 465

Related operations `list apps`

470 **4.1.13 list jobs**

Functional description This operation returns a list of jobs. This operation can be executed either as a project admin, a system admin, a job provider, an application provider or as an data provider.

- 475 • Project admin: As a project admin, it returns a list of all jobs in your projects. If a project is given it returns the jobs in that specific project.
- System admin: As system admin, it returns a list of all jobs in the system. If a project is given it returns the jobs in that specific project.
- Job provider: As job provider, it returns a list of all your jobs. If a project is given it returns all your jobs in that specific project.
- 480 • Application provider: As application provider it returns a list of the jobs that use an application that you have provided. If a project is given it returns all jobs in the specific project that use an application that you have provided.
- Data provider: As data provider it returns a list of the jobs that use one or more datasets that you have provided. If a project is given it returns all jobs in the specific project that use one or more datasets that you have provided.
- 485

Formal description

- Syntax: `as rol:<rolename> list jobs [in pro:<projectname>]`
- Parameters:
 - 490 – `<rolename>` (Required) : This parameter is the name of the role you have. This can either be "projadmin", "sysadmin", "jobprov", "appprov" or "datprov".
 - `<projectname>` (Optional) : This parameter is the name of the project. See the functional description of this operation for more information.

Examples

- 495 • `as rol:jobprov list jobs`
- `as rol:sysadmin list jobs in pro:MedicalResearch`

Possible errors

- You do not have the role project admin.
- You do not have the role system admin.
- 500 • You do not have the role job provider.

- You do not have the role application provider.
- You do not have the role data provider.
- Project <projectname> does not exist.
- You do not have the role project admin in project <projectname>.
- 505 • You do not have the role job provider in project <projectname>.

Related operations None

4.1.14 list projects

Functional description This operation returns a list of projects. This operation can be executed either as a project admin, a system admin, a job provider, an application provider
510 or as an data provider.

- Project admin: As a project admin, it returns a list of all projects where you are project admin.
- System admin: As system admin, it returns a list of all projects in the system.
- Job provider: As job provider, it returns a list of all projects where you can submit jobs
515 to.
- Application provider: As application provider, it returns a list of all projects in the system.
- Data provider: As data provider, it returns a list of all projects in the system.

Formal description

- 520 • Syntax: as rol:<rolename> list projects
- Parameters:
 - <rolename> (Required) : This parameter is the name of the role you have. This can either be "projadmin", "sysadmin", "jobprov", "appprov" or "datprov".

525 Examples

- as rol:sysadmin list projects
- as rol:datprov list projects

Possible errors

- You do not have the role project admin.
- 530 • You do not have the role system admin.
- You do not have the role job provider.
- You do not have the role application provider.
- You do not have the role data provider.

Related operations535 **4.1.15 list users**

Functional description This operation returns a list of users having a specific role in a specific project.

Formal description

- 540 • Syntax: as rol:<rolename1> list users having rol:<rolename2> [in pro:<projectname>]
- Parameters:
 - <rolename1> (Required) : This parameter is the name of the role you have. This parameter can only be "projadmin" or "sysadmin".
 - 545 – <rolename> (Required) : This parameter is the name of the role the users have in the returned list.
 - <projectname> (Optional) : This parameter is the name of the project which the users in the returned list are in.

Examples

- as rol:sysadmin list users having rol:datprov
- 550 • as rol:projadmin list users having rol:jobprov in pro:MedicalResearch

Possible errors

- You do not have any of the following roles: project admin, system admin.
- You do not have the role project admin.
- You do not have the role system admin.
- 555 • Project <projectname> does not exist.

Related operations None

4.1.16 passwd

Functional description This operation changes your password.

Formal description

- 560
- Syntax: `passwd <password>`
 - Parameters:
 - `<password>` (Required) : This is the new password where the old password is changed into. Spaces are not allowed.

565 Examples

- `passwd teiP`

Possible errors None

Related operations None

4.1.17 remove

570 **Functional description** This operation removes an application, a dataset, a project, a job or an user from the SPINGRID system.

Formal description

- 575
- Syntax: `remove (app:<applicationname> | dat:<datasetname> | pro:<projectname> | job:<jobname> | usr:<username>)`
 - Parameters:
 - `<applicationname>` (Required when chosen) : This parameter is the name of the application that is removed from the SPINGRID system.
 - `<datasetname>` (Required when chosen) : This parameter is the name of the dataset that is removed from the SPINGRID system.
 - `<projectname>` (Required when chosen) : This parameter is the name of the project that is removed from the SPINGRID system.
- 580

- `<jobname>` (Required when chosen) : This parameter is the name of the job that is removed from the SPINGRID system.
- 585 – `<username>` (Required when chosen) : This parameter is the name of the user that is removed from the SPINGRID system.

Examples

- `remove pro:MedicalResearch`
- `remove usr:Henk`

590 Possible errors

- You do not have the role project admin.
- You do not have the role system admin.
- User `<username>` does not exist.
- 595 • User `<username>` is the last project admin of project `<projectname>`. You cannot remove this user.
- User `<username>` does not have the role project admin in project `<projectname>`.

Related operations

4.1.18 trust

600 **Functional description** This operation lets an user trust an application, dataset or user for a project.

Formal description

- Syntax: `trust (app:<applicationname> | <datasetname> | <username> as rol:<rolename>) [for <projectname>]`

605 • Parameters:

- `<applicationname>` (Required when chosen) : This parameter is the name of the application that is trusted.
- `<datasetname>` (Required when chosen) : This parameter is the name of the dataset that is trusted.
- 610 – `<username>` (Required when chosen) : This parameter is the name of the user that is trusted.

- `<rolename>` (Required when chosen) : This parameter is the name of the role of the user that is trusted.
- `<projectname>` (Required when chosen) : This parameter is the name of the project where the user is trusted in. This parameter is required when trusting an application or data provider.

Examples

Possible errors

- User `<username>` does not exist.
- User `<username>` is already a job provider in project `<projectname>`.
- Project `<projectname>` does not exist.
- You do not have the role application provider.
- You do not have the role data provider.
- You do not have the role project admin in project `<projectname>`.
- DataSet `<datasetname>` does not exist.
- Application `<applicationname>` does not exist.
- DataSet `<datasetname>` is not yours.
- Application `<applicationname>` is not yours.
- You, as application provider already allowed project `<projectname>` to use your application `<applicationname>`.
- You, as data provider already allowed project `<projectname>` to use your dataset `<datasetname>`.
- Project `<projectname>` does not exist.

Related operations `distrust`

4.1.19 `unassign`

Functional description This operation removes a role from an user.

Formal description

- Syntax: `unassign rol:<rolename> from usr:<username> [in pro:<projectname>]`

- 640 • Parameters:

- `<rolename>` (Required) : This parameter is the name of role that is removed from the user.
- `<username>` (Required) : This parameter is the name of the user where the role is removed from.
- 645 – `<projectname>` (Optional) : This parameter is the name of the project where the user as the role is removed from. This parameter is only necessary when removing the role project admin or job provider from an user.

Examples

- `unassign rol:datprov from usr:Henk`
- 650 • `unassign rol:projadmin from usr:Piet in pro:MedicalResearch`

Possible errors

- You do not have the role system admin.
- User `<username>` does not exist.
- Project `<projectname>` does not exist.
- 655 • You do not have the role of project admin in project `<projectname>`.
- User `<username>` does not have the role of application provider.
- User `<username>` does not have the role of data provider.
- User `<username>` does not have the role of job provider in project `<projectname>`.
- User `<username>` does not have the role of project admin in project `<projectname>`.

- 660 **Related operations** `assign`

4.2 Agent operations**4.2.1 add interval**

Functional description This operation adds an interval when the resource can be used to execute jobs.

665 Formal description

- Syntax: add interval tim:<time1> to tim:<time2>
- Parameters:
 - <time1> (Required) : This parameter is the first parameter of the interval that is added. The format is *hhmmss*.
 - <time2> (Required) : This parameter is the second parameter of the interval that is added. The format is *hhmmss*.

Examples

- add interval tim:140000 to tim tim:180000

675 Possible errors None**Related operations** remove interval**4.2.2 exit****Functional description** This operation exits the agent application.**Formal description**

- 680**
- Syntax: exit

Examples

- exit

Possible errors None**685 Related operations** None**4.2.3 help****Functional description** This operation returns information of a command.

Formal description

- Syntax: list help <command>

690

- Parameters:

- <command> (Required) : This parameter is name of the command where information of is returned.

Examples

695

- help add interval

Possible errors None**Related operations** None**4.2.4 list projects**

700

Functional description This operation returns a list of all projects in the SPINGRID system.**Formal description**

- Syntax: list projects

Examples

705

- list projects

Possible errors None**Related operations** None**4.2.5 remove interval**

710

Functional description This operations removes an interval when the resource can be used to execute jobs.

Formal description

- Syntax: remove interval tim:<time1> to tim:<time2>
- Parameters:

- 715
- <time1> (Required) : This parameter is the first parameter of the interval that is removed. The format is *hhmmss*.
 - <time2> (Required) : This parameter is the second parameter of the interval that is removed. The format is *hhmmss*.

Examples

- 720
- remove interval tim:140000 to tim tim:180000

Possible errors None

Related operations add interval

4.3 Dispatcher operations

4.3.1 exit

- 725 **Functional description** This operation exits the dispatcher. The dispatcher stops running after calling the operation.

Formal description

- Syntax: exit

730 **Examples**

- exit

Possible errors None

Related operations None

Appendix A

⁷³⁵ Error messages and recovery procedures

All error messages can be found in chapter 4. There are no recovery procedures if an error occurs. If the application crashes then the program should be restarted as described in chapter 3.

Appendix B

Glossary

Agent	Application that is used by a resource provider to retrieve and execute jobs.
Application	A non-interactive data processing application consisting of executables, scripts and/or auxiliary data files that reads one or more input data files and writes one ore more output files.
Application Provider	An application provider can offer a set of applications to the SPINGRID system. They can restrict access for projects and for resource providers to their applications.
Client	Application that is used by all the users except the resource provider who uses the agent application.
Computational Grid	A hardware and software infrastructure that enables coordinated resource sharing within dynamic organizations consisting of individuals, institutions and resources.
Customer	Dutch Space B.V.
Data Provider	A data provider can offer a set of datafiles to the SPINGRID system. They can restrict access for projects and for resource providers to their datafiles.
Dispatcher	A dispatcher acts like a server and manages the distribution of jobs over the computational grid.
Job	Specification of application, configuration data, input and/or output data files and scheduler specific data (priority, preferred resource, etc).
Job Provider	Job providers are users that offer a job to a project. They have to be a member of that particular project.
Project	A collection of jobs with specified access rights to which users (project members) can be assigned.
Project Administrator	The project administrators administrate projects and can assign and remove job providers, configure a project and restrict access for resource providers.
Resource Provider	Resource providers are users that offer time on their computers to the SPINGRID system. They can restrict access to their computer for application providers and projects.
Role	The actions and activities assigned to a person.

SPINGRID	A computational grid using SPINGRID software.
SPINGRID Software	Software developed by Dutch Space and TU/e to build computational grids for distributed data processing.
SPINGRID System	The full name of the entire system.
System Administrator	The system administrator oversees the entire SPINGRID system and has the right to configure the system, to create and remove projects and assign and remove project administrators.

Index

agent

- add interval, 31
- exit, 32
- 745 help, 32
- list projects, 33
- remove interval, 33

client

- add app, 13
- 750 add data, 14
- add job, 15
- add project, 16
- add user, 16
- allow private, 17
- 755 assign, 18
- disallow private, 19
- distrust, 20
- info, 21
- list apps, 22
- 760 list data, 23
- list jobs, 25
- list projects, 26
- list users, 27
- passwd, 28
- 765 remove, 28
- trust, 29
- unassign, 30

dispatcher

- exit, 34