

Software Engineering Project (2IP40)

Project Group 1

Detailed Design Document

version 0.0.2 (Draft), 12 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:	M. ter Linden	Dutch Space
	H. de Wolf	Dutch Space

Abstract

This document contains the Detailed Design for the SPINGRID system. This project is one of seven assignments for the course 2IP40 Software Engineering at Eindhoven University of Technology. It defines the design and coding standards. Furthermore it describes the detailed design using the software requirements and the logical model defined in the [SRD] and the component model defined in the [ADD]. The document complies with the Detailed Design Document (DDD) from the Software Engineering Standard, as set by the European Space Agency [ESA].

Contents

1	Introduction	7
1.1	Purpose	7
1.2	Scope	7
1.3	List of definitions and abbreviations	8
1.3.1	Definitions	8
1.3.2	Abbreviations	9
1.4	Documents	9
1.4.1	Reference Documents	9
1.4.2	Applicable Documents	9
1.5	Overview	10
2	Standards and conventions	11
2.1	Design standards	11
2.2	Documentation standards	11
2.3	Naming conventions	11
2.4	Coding standards	11
2.5	Software development tools	11
3	Component descriptions	12
3.1	Package dClientManager	13
3.1.1	Classes	14
3.2	Package dIOProcessor	33
3.2.1	Classes	34
3.3	Package aIOProcessor	37
3.3.1	Classes	38

3.4	Package dAgentManager	44
3.4.1	Classes	45
3.5	Package aDistributor	48
3.5.1	Classes	49
3.6	Package commands	51
3.6.1	Classes	52
3.7	Package interfaces	58
3.7.1	Interfaces	61
3.8	Package cTranslator	119
3.8.1	Classes	120
3.9	Package aJobExecutor	126
3.9.1	Classes	127
3.10	Package dDatabaseManager	129
3.10.1	Classes	130
3.11	Package JSDL	132
3.11.1	Classes	133
3.12	Package types	163
3.12.1	Classes	164
3.13	Package configuration	172
3.13.1	Classes	173
3.14	Package aTranslator	175
3.14.1	Classes	176
3.15	Package dTranslator	178
3.15.1	Classes	179
3.16	Package cLogger	181
3.16.1	Classes	182
3.17	Package aLogger	185
3.17.1	Classes	186
3.18	Package aDataManager	189
3.18.1	Interfaces	190
3.18.2	Classes	190
3.19	Package genericpackages	202

3.19.1	Classes	203
3.20	Package aJobScheduler	211
3.20.1	Classes	212
3.21	Package exceptions	215
3.21.1	Classes	216
3.22	Package cCommunicator	244
3.22.1	Classes	245
3.23	Package HTTPMessage	246
3.23.1	Classes	247
3.24	Package cIOProcessor	253
3.24.1	Classes	254
3.25	Package aCommunicator	263
3.25.1	Classes	264
3.26	Package dLogger	265
3.26.1	Classes	266
3.27	Package dCommunicator	269
3.27.1	Classes	270
3.28	Package <none>	272
3.28.1	Classes	273
4	Build procedure	275
A	Source code listings	276
A.1	Client Components	276
A.2	Agent Components	277
A.3	Dispatcher Components	278
B	Requirements traceability matrix	281

Document Status Sheet

Document Title	Detailed Design Document
Document Identification	SPINGRID/Documents/Product/DDD/0.0.2
Author(s)	R. Leeuwestein
Version	0.0.2
Document Status	<u>draft</u> / internally accepted / conditionally approved / approved

Version	Date	Author(s)	Summary
0.0.1	27-04-2006	R. Leeuwestein	Document creation
0.0.2	12-06-2006	R. Leeuwestein	Draft

Document Change Report

Document Title	Detailed Design Document
Document Identification	SPINGRID/Documents/Product/DDD/0.0.2
Date of Changes	N/A

Section Number	Reason for Change
-----------------------	--------------------------

Chapter 1

Introduction

1.1 Purpose

The purpose of the Detailed Design Document (DDD) is to define the detailed design for
5 all components of the SPINGRID system which are specified in the [SRD] and [ADD]. The
low-level components are designed, coded and tested.

1.2 Scope

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

1.3 List of definitions and abbreviations

1.3.1 Definitions

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 or 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.
Dispatcher	A dispatcher acts like a server and manages the distribution of jobs over the computational grid.
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.
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.
GRP	A generic package that is converted from a HTTP response and is formatted as two strings (status code and message).
GSP	A generic package that can be converted to a HTTP request header and is formatted as a list of pairs of two strings (variable and value).
PM	Project Manager.
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.

1.3.2 Abbreviations

ADD	Architectural Design Document
ESA	European Space Agency
DDD	Detailed Design Document
GRP	Generic Receive Package
GSP	Generic Send Package
HTTP	Hypertext Transfer Protocol
JRE	Java Runtime Environment
OMT	Object Modeling Technique
SRD	Software Requirements Document
UML	Unified Modeling Language

15

1.4 Documents

1.4.1 Reference Documents

[ESA]	<i>ESA Software Engineering Standards (ESA PSS-05-0 Issue 2)</i> , ESA Board for Software Standardization and Control (BSSC), 1991
[BSSC]	<i>BSSC 2005 Java Coding Standard</i> , Issue 1.0, March 2005
[JSDL]	<i>Job Submission Description Language (JSDL) Specification</i> , Version 1.0, November 2005
[SUM]	<i>Software User Manual</i> , SPINGRID team, TU/e, Version 0.0.2, June 2006
[SCMP]	<i>Software Configuration Management Plan</i> , SPINGRID team, TU/e, Version 0.1.1, February 2006
[SQAP]	<i>Software Quality Assurance Plan</i> , SPINGRID team, TU/e, Version 0.1.2, March 2006
[SVVP]	<i>Software Verification and Validation Plan</i> , SPINGRID team, TU/e, Version 0.1.2, March 2006
[UML]	<i>Practical UML: A Hands-On Introduction for Developers</i> , Togethersoft, 2000

1.4.2 Applicable Documents

[ADD]	<i>Architectural Design Document</i> , SPINGRID team, TU/e, version 1.0.0, 27 April 2006
[SRD]	<i>Software Requirements Document</i> , SPINGRID team, TU/e, version 1.0.1, March 2006
[URD]	<i>User Requirements Document</i> , SPINGRID team, TU/e, version 1.0.0, February 2006

1.5 Overview

20 This DDD is organized as follows. Chapter 2 of this document is a short introduction to the general context of the system to be made and to the background of this project. Chapter 3 describes all components in a consistent way. This chapter is generated with javadoc (<http://java.sun.com/j2se/javadoc/>). Chapter 4 contains a reference to where the build procedure of the SPINGRID system can be found. Chapter 5 lists the files in which the components
25 reside. Chapter 6 contains the traceability matrix.

Chapter 2

Standards and conventions

2.1 Design standards

The method used to design the component model is UML. The presented model is a decomposition view into components (refer to [ADD, chapter 4]).

2.2 Documentation standards

The documentation standards are described in [SQAP, section 4.1].

2.3 Naming conventions

The naming conventions are described in [SCMP, section 3.1].

2.4 Coding standards

The coding standards are given in [BSSC].

2.5 Software development tools

The software development tools are described in [SCMP, chapter 6]

Chapter 3

40 Component descriptions

3.1 Package dClientManager

Package Contents *Page*

Classes

DClientManager 14

*The Client Manager has several functions: - Receive the commands from the
Client Translator or the IOProcessor.*

EMessages 21

...no description...

QueryCreator 26

...no description...

45

3.1.1 Classes

50 CLASS **DClientManager**

The Client Manager has several functions: - Receive the commands from the Client Translator or the IOProcessor. - Perform the desired actions. - Update/Read the database. - Send the requested results back to the Client Translator or the IOProcessor.

55 DECLARATION

```
public class DClientManager

extends java.lang.Object
implements interfaces.IDCCommandInput, interfaces.IDUserCommands,
interfaces.IDProjectAdminCommands, interfaces.IDSystemAdminCommands,
interfaces.IDJobProviderCommands, interfaces.IDApplicationProviderCommands,
interfaces.IDDataProviderCommands
```

CONSTRUCTORS

60

- *DClientManager*
public **DClientManager**()
 - **Usage**
 - * Creates a new instance of this component

65 METHODS

- *AddApplication*
public void **AddApplication**(java.lang.String user, java.lang.String hash, java.lang.String applicationxml, java.lang.String applicationname, java.lang.String description)
- *AddApplicationProvider*
public void **AddApplicationProvider**(java.lang.String user, java.lang.String hash, java.lang.String username)
- *AddData*
public void **AddData**(java.lang.String user, java.lang.String hash, java.net.URL [] data, java.lang.String dataname, java.lang.String description)

80

- *AddDataProvider*
`public void AddDataProvider(java.lang.String user, java.lang.String hash, java.lang.String username)`

- 85 • *AddProject*
`public void AddProject(java.lang.String user, java.lang.String hash, java.lang.String projectname, java.lang.String username)`

- *AddProjectAdmin*
90 `public void AddProjectAdmin(java.lang.String user, java.lang.String hash, java.lang.String username, java.lang.String projectname)`

- *AddUser*
`public void AddUser(java.lang.String user, java.lang.String hash, java.lang.String username, java.lang.String password)`
95

- *AllowOwnApplication*
`public void AllowOwnApplication(java.lang.String user, java.lang.String hash, java.lang.String projectname, java.lang.String jobprovider)`
100

- *AllowOwnData*
`public void AllowOwnData(java.lang.String user, java.lang.String hash, java.lang.String projectname, java.lang.String jobprovider)`
105

- *ApproveApplicationProvider*
`public void ApproveApplicationProvider(java.lang.String user, java.lang.String hash, java.lang.String projectname, java.lang.String applicationprovider)`
110

- *ApproveApplicationToProject*
`public void ApproveApplicationToProject(java.lang.String user, java.lang.String hash, java.lang.String application, java.lang.String projectname)`
115

- *ApproveDataProvider*
`public void ApproveDataProvider(java.lang.String user, java.lang.String hash, java.lang.String projectname, java.lang.String dataprovider)`
120

- *ApproveDataToProject*
`public void ApproveDataToProject(java.lang.String user, java.lang.String hash, java.lang.String data, java.lang.String projectname)`
125

- *ApproveJobProviderForProjectAdmin*
`public void ApproveJobProviderForProjectAdmin(java.lang.String user, java.lang.String hash, java.lang.String projectname, java.lang.String jobprovider)`
130

- *begin*
`public void begin()`

– Usage

* Method that is called when an instance has been created and the interfaces have been set

• *DisallowOwnApplication*

```
public void DisallowOwnApplication( java.lang.String user,
java.lang.String hash, java.lang.String projectname,
java.lang.String jobprovider )
```

• *DisallowOwnData*

```
public void DisallowOwnData( java.lang.String user, java.lang.String
hash, java.lang.String projectname, java.lang.String jobprovider )
```

• *DisapproveApplicationProvider*

```
public void DisapproveApplicationProvider( java.lang.String user,
java.lang.String hash, java.lang.String projectname,
java.lang.String applicationprovider )
```

• *DisapproveApplicationToProject*

```
public void DisapproveApplicationToProject( java.lang.String user,
java.lang.String hash, java.lang.String application, java.lang.String
projectname )
```

• *DisapproveDataProvider*

```
public void DisapproveDataProvider( java.lang.String user,
java.lang.String hash, java.lang.String projectname,
java.lang.String dataprovider )
```

• *DisapproveDataToProject*

```
public void DisapproveDataToProject( java.lang.String user,
java.lang.String hash, java.lang.String data, java.lang.String
projectname )
```

• *DisapproveJobProviderForProjectAdmin*

```
public void DisapproveJobProviderForProjectAdmin( java.lang.String
user, java.lang.String hash, java.lang.String projectname,
java.lang.String jobprovider )
```

• *infoApplication*

```
public String infoApplication( java.lang.String user, java.lang.String
hash, java.lang.String applicationname )
```

• *infoDataSet*

```
public String infoDataSet( java.lang.String user, java.lang.String
hash, java.lang.String datasetname )
```

• *infoJob*

```
public String infoJob( java.lang.String user, java.lang.String hash,
java.lang.String jobname )
```

```

185 • infoProject
    public String infoProject( java.lang.String user, java.lang.String
    hash, java.lang.String projectname )

190 • infoUser
    public String infoUser( java.lang.String user, java.lang.String hash,
    java.lang.String username )

    • listApplicationProvidersForProjectAdmin
    public String listApplicationProvidersForProjectAdmin( java.lang.String
    user, java.lang.String hash )

195 • listApplicationProvidersForSystemAdmin
    public String listApplicationProvidersForSystemAdmin( java.lang.String
    user, java.lang.String hash )

    • listApplicationsForApplicationProvider
200 public String listApplicationsForApplicationProvider( java.lang.String
    user, java.lang.String hash )

    • listApplicationsForJobProvider
    public String listApplicationsForJobProvider( java.lang.String user,
205 java.lang.String hash )

    • listApplicationsForJobProvider
    public String listApplicationsForJobProvider( java.lang.String user,
210 java.lang.String hash, java.lang.String project )

    • listApplicationsForProjectAdmin
    public String listApplicationsForProjectAdmin( java.lang.String user,
    java.lang.String hash )

215 • listApplicationsForProjectAdmin
    public String listApplicationsForProjectAdmin( java.lang.String user,
    java.lang.String hash, java.lang.String project )

    • listApplicationsForSystemAdmin
220 public String listApplicationsForSystemAdmin( java.lang.String user,
    java.lang.String hash )

    • listApplicationsForSystemAdmin
    public String listApplicationsForSystemAdmin( java.lang.String user,
225 java.lang.String hash, java.lang.String project )

    • listDataProvidersForProjectAdmin
    public String listDataProvidersForProjectAdmin( java.lang.String
230 user, java.lang.String hash )

    • listDataProvidersForSystemAdmin
    public String listDataProvidersForSystemAdmin( java.lang.String
    user, java.lang.String hash )

```

```

235 • listDataSetsForDataProvider
    public String listDataSetsForDataProvider( java.lang.String user,
    java.lang.String hash )
    _____
240 • listDataSetsForJobProvider
    public String listDataSetsForJobProvider( java.lang.String user,
    java.lang.String hash )
    _____
245 • listDataSetsForJobProvider
    public String listDataSetsForJobProvider( java.lang.String user,
    java.lang.String hash, java.lang.String project )
    _____
    • listDataSetsForProjectAdmin
    public String listDataSetsForProjectAdmin( java.lang.String user,
    java.lang.String hash )
250 _____
    • listDataSetsForProjectAdmin
    public String listDataSetsForProjectAdmin( java.lang.String user,
    java.lang.String hash, java.lang.String project )
    _____
255 • listDataSetsForSystemAdmin
    public String listDataSetsForSystemAdmin( java.lang.String user,
    java.lang.String hash )
    _____
    • listDataSetsForSystemAdmin
260 public String listDataSetsForSystemAdmin( java.lang.String user,
    java.lang.String hash, java.lang.String project )
    _____
    • listJobProvidersForProjectAdmin
    public String listJobProvidersForProjectAdmin( java.lang.String user,
265 java.lang.String hash )
    _____
    • listJobProvidersForProjectAdmin
    public String listJobProvidersForProjectAdmin( java.lang.String user,
    java.lang.String hash, java.lang.String project )
270 _____
    • listJobProvidersForSystemAdmin
    public String listJobProvidersForSystemAdmin( java.lang.String user,
    java.lang.String hash )
    _____
275 • listJobProvidersForSystemAdmin
    public String listJobProvidersForSystemAdmin( java.lang.String user,
    java.lang.String hash, java.lang.String project )
    _____
    • listJobsForJobProvider
280 public String listJobsForJobProvider( java.lang.String user,
    java.lang.String hash )
    _____
    • listJobsForJobProvider
    public String listJobsForJobProvider( java.lang.String user,
285 java.lang.String hash, java.lang.String project )
    _____

```

- *listJobsForProjectAdmin*
 public String **listJobsForProjectAdmin**(java.lang.String user,
 java.lang.String hash)

- *listJobsForProjectAdmin*
 public String **listJobsForProjectAdmin**(java.lang.String user,
 java.lang.String hash, java.lang.String project)

- *listJobsForSystemAdmin*
 public String **listJobsForSystemAdmin**(java.lang.String user,
 java.lang.String hash)

- *listJobsForSystemAdmin*
 public String **listJobsForSystemAdmin**(java.lang.String user,
 java.lang.String hash, java.lang.String project)

- *listJobsThatUseAppForApplicationProvider*
 public String **listJobsThatUseAppForApplicationProvider**(
 java.lang.String user, java.lang.String hash)

- *listJobsThatUseAppForApplicationProvider*
 public String **listJobsThatUseAppForApplicationProvider**(
 java.lang.String user, java.lang.String hash, java.lang.String
 project)

- *listJobsThatUseDataForDataProvider*
 public String **listJobsThatUseDataForDataProvider**(java.lang.String
 user, java.lang.String hash)

- *listJobsThatUseDataForDataProvider*
 public String **listJobsThatUseDataForDataProvider**(java.lang.String
 user, java.lang.String hash, java.lang.String project)

- *listProjectAdminsForSystemAdmin*
 public String **listProjectAdminsForSystemAdmin**(java.lang.String
 user, java.lang.String hash)

- *listProjectsForApplicationProvider*
 public String **listProjectsForApplicationProvider**(java.lang.String
 user, java.lang.String hash)

- *listProjectsForDataProvider*
 public String **listProjectsForDataProvider**(java.lang.String user,
 java.lang.String hash)

- *listProjectsForJobProvider*
 public String **listProjectsForJobProvider**(java.lang.String user,
 java.lang.String hash)

- *listProjectsForProjectAdmin*
 public String **listProjectsForProjectAdmin**(java.lang.String user,
 java.lang.String hash)

340 • *listProjectsForSystemAdmin*
 public String **listProjectsForSystemAdmin**(java.lang.String user,
 java.lang.String hash)

345 • *listUsers*
 public String **listUsers**(java.lang.String user, java.lang.String hash
)

350 • *OfferJob*
 public void **OfferJob**(java.lang.String user, java.lang.String hash,
 java.lang.String projectname, java.lang.String job)

355 • *RemoveApplication*
 public void **RemoveApplication**(java.lang.String user,
 java.lang.String hash, java.lang.String application)

360 • *RemoveApplicationProvider*
 public void **RemoveApplicationProvider**(java.lang.String user,
 java.lang.String hash, java.lang.String username)

365 • *RemoveData*
 public void **RemoveData**(java.lang.String user, java.lang.String
 hash, java.lang.String data)

370 • *RemoveDataProvider*
 public void **RemoveDataProvider**(java.lang.String user,
 java.lang.String hash, java.lang.String username)

375 • *RemoveJob*
 public void **RemoveJob**(java.lang.String user, java.lang.String
 hash, java.lang.String job)

380 • *RemoveJobForJobProvider*
 public void **RemoveJobForJobProvider**(java.lang.String user,
 java.lang.String hash, java.lang.String job)

385 • *RemoveProject*
 public void **RemoveProject**(java.lang.String user, java.lang.String
 hash, java.lang.String projectname)

390 • *RemoveProjectAdmin*
 public void **RemoveProjectAdmin**(java.lang.String user,
 java.lang.String hash, java.lang.String username, java.lang.String
 projectname)

390 • *RemoveUserForSystemAdmin*
 public void **RemoveUserForSystemAdmin**(java.lang.String user,
 java.lang.String hash, java.lang.String username)

390 • *setInterfaces*
 public void **setInterfaces**(interfaces.IDQueryHandler idQueryHandler,
 interfaces.IDOutputProcessor idOutputProcessor)

- **Usage**

- * Sets references to all interfaces this component might need

- **Parameters**

- * `idQueryHandler` -
 - * `idOutputProcessor` -

- *SetPassword*

```
public void SetPassword( java.lang.String username, java.lang.String  
hash, java.lang.String newpassword )
```

- *Shutdown*

```
public void Shutdown( )
```

CLASS **EMessages**

DECLARATION

```
public class EMessages  
extends java.lang.Object
```

FIELDS

- `public static final String ROLE_APPPROV`
 - Constante representing application provider role
- `public static final String ROLE_DATAPROV`
 - Constante representing data provider role
- `public static final String ROLE_JOBPROV`
 - Constante representing job provider role
- `public static final String ROLE_PROJADMIN`
 - Constante representing job provider role
- `public static final String ROLE_SYSADMIN`
 - Constante representing system admin role

CONSTRUCTORS

- *EMessages*

```
public EMessages( )
```

METHODS

• *alreadyRole*

```
public static String alreadyRole( java.lang.String user,  
430 java.lang.String role )
```

– Usage

* Generates an error message

– Parameters

* **user** - username

435 * **role** - role-string

– Returns - Error message

• *alreadyRoleInProject*

```
public static String alreadyRoleInProject( java.lang.String user,  
440 java.lang.String role, java.lang.String project )
```

– Usage

* Generates an error message

– Parameters

* **user** - username

445 * **role** - role-string

– Returns - Error message

• *alreadyTrustUserInProject*

```
public static String alreadyTrustUserInProject( java.lang.String  
450 roleA, java.lang.String project, java.lang.String userB,  
java.lang.String roleB )
```

– Usage

* Generates an error message

– Parameters

455 * **user** - username

* **role** - role-string

– Returns - Error message

• *alreadyTrustUserInProject2*

```
public static String alreadyTrustUserInProject2( java.lang.String  
460 roleA, java.lang.String project, java.lang.String userB,  
java.lang.String roleB )
```

– Usage

* Generates an error message

– Parameters

465 * **user** - username

* **role** - role-string

– **Returns** - Error message

• *applicationAlreadyExists*

```
public static String applicationAlreadyExists( java.lang.String  
application )
```

– **Usage**

* Generates an error message

– **Parameters**

* **user** - username

* **role** - role-string

– **Returns** - Error message

• *applicationNotExists*

```
public static String applicationNotExists( java.lang.String application  
)
```

– **Usage**

* Generates an error message

– **Parameters**

* **user** - username

* **role** - role-string

– **Returns** - Error message

• *dataAlreadyExists*

```
public static String dataAlreadyExists( java.lang.String application )
```

– **Usage**

* Generates an error message

– **Parameters**

* **user** - username

* **role** - role-string

– **Returns** - Error message

• *dataNotExists*

```
public static String dataNotExists( java.lang.String data )
```

– **Usage**

* Generates an error message

– **Parameters**

* **user** - username

* **role** - role-string

– **Returns** - Error message

• *jobNotExists*

```
public static String jobNotExists( java.lang.String job )
```

– **Usage**


```
        * Generates an error message
    - Parameters
        * user - username
        * role - role-string
515    - Returns - Error message
    _____
    • noRole
      public static String noRole( java.lang.String  role )

        - Usage
520        * Generates an error message
        - Parameters
            * user - username
            * role - role-string
        - Returns - Error message
525    _____
    • noRoleInProject
      public static String noRoleInProject( java.lang.String  role,
      java.lang.String  project )

        - Usage
530        * Generates an error message
        - Parameters
            * user - username
            * role - role-string
        - Returns - Error message
535    _____
    • noTrustInProject
      public static String noTrustInProject( java.lang.String  roleA,
      java.lang.String  project, java.lang.String  userB, java.lang.String
      roleB )

540        - Usage
            * Generates an error message
        - Parameters
            * user - username
            * role - role-string
545        - Returns - Error message
    _____
    • noTrustInProject2
      public static String noTrustInProject2( java.lang.String  roleA,
      java.lang.String  project, java.lang.String  userB, java.lang.String
550      roleB )

        - Usage
            * Generates an error message
        - Parameters
```

```

555         * user - username
        * role - role-string
        – Returns - Error message
        _____
    • projectAlreadyExists
      public static String projectAlreadyExists( java.lang.String project )

560        – Usage
          * Generates an error message
        – Parameters
          * user - username
          * role - role-string
565        – Returns - Error message
        _____
    • projectNotExists
      public static String projectNotExists( java.lang.String project )

        – Usage
570        * Generates an error message
        – Parameters
          * user - username
          * role - role-string
        – Returns - Error message
575        _____
    • userExists
      public static String userExists( java.lang.String user )

        – Usage
          * Generates an error message
580        – Parameters
          * user - username
          * role - role-string
        – Returns - Error message
        _____
585    • userNoRole
      public static String userNoRole( java.lang.String user,
      java.lang.String role )

        – Usage
          * Generates an error message
590        – Parameters
          * user - username
          * role - role-string
        – Returns - Error message
        _____
595    • userNoRoleInProject
      public static String userNoRoleInProject( java.lang.String user,
      java.lang.String role, java.lang.String project )

```

- **Usage**
 - * Generates an error message

600

- **Parameters**

- * **user** - username
- * **role** - role-string

- **Returns** - Error message
-

605

- *userNotExists*

```
public static String userNotExists( java.lang.String user )
```

- **Usage**
 - * Generates an error message

610

- **Parameters**

- * **user** - username
- * **role** - role-string

- **Returns** - Error message

CLASS **QueryCreator**

615

DECLARATION

```
public class QueryCreator
extends java.lang.Object
```

FIELDS

620

- public static final int SYSTEMADMINISTRATOR
 - Constant representing a role
- public static final int SYSADMIN
 - Constant representing a role
- 625 • public static final int APPLICATIONPROVIDER
 - Constant representing a role
- public static final int APPPROV
 - Constant representing a role
- public static final int DATAPROVIDER
 - 630 – Constant representing a role

- public static final int DATAPROV
 - Constant representing a role
- public static final int PROJECTADMINISTRATOR
 - Constant representing a role
- 635 • public static final int PROJADMIN
 - Constant representing a role
- public static final int JOBPROVIDER
 - Constant representing a role
- public static final int JOBPROV
 - 640 – Constant representing a role
- public static final int PRIVATEAPPLICATIONPROVIDER
 - Constant representing a role
- public static final int PRIVAPPPROV
 - Constant representing a role
- 645 • public static final int PRIVATEDATAPROVIDER
 - Constant representing a role
- public static final int PRIVDATAPROV
 - Constant representing a role

CONSTRUCTORS

- 650
- *QueryCreator*
 public **QueryCreator**()

METHODS

- 655
- *ANowDistrustsB*
 public static String **ANowDistrustsB**(java.lang.String **AID**,
 java.lang.String **BID**)
 – **Usage**
 * Create query let one HasRole instance distrust another
 – **Parameters**
 * **AID** - HasRoleID
 * **BID** - HasRoleID
 – **Returns** - A query
- 660

- 665 • *ANowTrustsB*
 public static String ANowTrustsB(java.lang.String AID,
 java.lang.String BID)
 – **Usage**
 * Create query let one HasRole instance trust another
 – **Parameters**
 * AID - HasRoleID
 * BID - HasRoleID
 – **Returns** - A query
- 670

- 675 • *ClearHasRoles*
 public static String ClearHasRoles(java.lang.String userID)

- *ClearTrust*
 public static String ClearTrust(java.lang.String trustee)
 – **Usage**
 * Create query to remove all instances of trust with the specified trustee
 – **Parameters**
 * trustee - UserID, the trustee
 – **Returns** - A query
- 680

- 685 • *ClearTrust*
 public static String ClearTrust(java.lang.String trustee, int
 RoleType)
 – **Usage**
 * Create query to remove all instances of trust with the specified trustee and
 role
 – **Parameters**
 * trustee - UserID, the trustee
 * roletype - Integers, use the constants in QueryCreator
 – **Returns** - A query
- 690

- 695 • *ClearTrust*
 public static String ClearTrust(java.lang.String trustee, int
 RoleType, java.lang.String projectID)
 – **Usage**
 * Create query to remove all instances of trust with the specified trustee and
 role in the specified project
 – **Parameters**
 * trustee - UserID, the trustee
 * roletype - Integers, use the constants in QueryCreator
 * projectID - ProjectID
- 700

- 705

– **Returns** - A query

- *ClearTrust2*

public static String **ClearTrust2**(java.lang.String **truster**)

– **Usage**

* Create query to remove all instances of trust with the specified truster

– **Parameters**

* **truster** - UserID, the trustee

– **Returns** - A query

- *DoesATrustB*

public static String **DoesATrustB**(java.lang.String **AID**,
java.lang.String **BID**)

– **Usage**

* Create query to check if one HasRole instance trusts another

– **Parameters**

* **AID** - HasRoleID

* **BID** - HasRoleID

– **Returns** - A query

- *GetApplicationID*

public static String **GetApplicationID**(java.lang.String
applicationname)

– **Usage**

* Create query to find the ApplicationID of the specified application

– **Parameters**

* **applicationname** - Name of the application

– **Returns** - A query

- *GetDataSetID*

public static String **GetDataSetID**(java.lang.String **datasetname**)

– **Usage**

* Create query to find the DataSetID of the specified dataset

– **Parameters**

* **datasetname** - Name of the dataset

– **Returns** - A query

- *GetHasRoleID*

public static String **GetHasRoleID**(java.lang.String **userID**, int
roletype)

– **Usage**

```

    * Create query to get the ID of the HasRole instances belonging to the
      specified parameters
750  - Parameters
      * userID - UserID
      * roletype - Integers, use the constants in QueryCreator
    - Returns - A query
  
```

```

755  • GetHasRoleID
    public static String GetHasRoleID( java.lang.String  userID, int
    roletype, java.lang.String  projectID )

    - Usage
      * Create query to get the ID of the HasRole instances belonging to the
760      specified parameters
    - Parameters
      * userID - UserID
      * roletype - Integers, use the constants in QueryCreator
      * projectID - ProjectID
765  - Returns - A query
  
```

```

    • GetJobID
    public static String GetJobID( java.lang.String  jobname )

    - Usage
770      * Create query to find the JobID of the specified job
    - Parameters
      * jobname - Name of the job
    - Returns - A query
  
```

```

775  • GetProjectID
    public static String GetProjectID( java.lang.String  projectname )

    - Usage
      * Create query to find the ProjectID of the specified project
    - Parameters
780      * projectname - Name of the project
    - Returns - A query
  
```

```

    • GetUser
    public static String GetUser( java.lang.String  username )

785  - Usage
      * Create query to get the password of the specified user
    - Parameters
      * username - Name of the user
    - Returns - A query
790
  
```

- *GetUserID*

```
public static String GetUserID( java.lang.String  username )
```

- **Usage**

- * Create query to find the UserID of the specified user

- **Parameters**

- * **username** - Name of the user

- **Returns** - A query

- *GiveRole*

```
public static String GiveRole( java.lang.String  userID, int  roletype  
)
```

- **Usage**

- * Create query to give a role to a user

- **Parameters**

- * **userID** - UserID

- * **roletype** - Integers, use the constants in QueryCreator

- **Returns** - A query

- *GiveRole*

```
public static String GiveRole( java.lang.String  userID, int  roletype,  
java.lang.String  projectID )
```

- **Usage**

- * Create query to give a role to a user

- **Parameters**

- * **userID** - UserID

- * **roletype** - Integers, use the constants in QueryCreator

- * **projectID** - ProjectID

- **Returns** - A query

- *TakeRole*

```
public static String TakeRole( java.lang.String  userID, int  roletype  
)
```

- **Usage**

- * Create query to take a role from a user

- **Parameters**

- * **userID** - UserID

- * **roletype** - Integers, use the constants in QueryCreator

- **Returns** - A query

- *TakeRole*

```
public static String TakeRole( java.lang.String  userID, int  roletype,  
java.lang.String  projectID )
```

- **Usage**

835

- * Create query to take a role from a user
- **Parameters**
 - * `userID` - UserID
 - * `roletype` - Integers, use the constants in QueryCreator
 - * `projectID` - ProjectID
- **Returns** - A query

3.2 Package dIOProcessor

Package Contents Page

Classes	
CommandParseErrors.....	34
<i>...no description...</i>	
CommandParser.....	34
<i>Class containing all error messages and descriptions that can be outputted by the CommandLineParser</i>	
dIOProcessor.....	35
<i>This component has the function to process inputs from the outside world and process outputs from the other components of the dispatcher.</i>	

3.2.1 Classes

CLASS **CommandParseErrors**

850

DECLARATION

```
public class CommandParseErrors
extends java.lang.Object
```

855

FIELDS

- `public static final String ERR_COMMANDS`
 - An error message that should be put when an invalid command was entered.
- `public static final String ERR_SYNTAX_EXIT`
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.

860

CONSTRUCTORS

- *CommandParseErrors*
`public CommandParseErrors()`

865

CLASS **CommandParser**

Class containing all error messages and descriptions that can be outputted by the
CommandLineParser

870

DECLARATION

```
public class CommandParser
extends java.lang.Object
```

CONSTRUCTORS

875

- *CommandParser*

```
public CommandParser( interfaces.IDCCommandInput  
idCCommandInput, interfaces.IDACommandInput idACommandInput,  
java.lang.String [] words )
```

- **Usage**

- * Constructs a new instance of the CommandParser

- **Parameters**

- * **words** - The sentence (words) on which the CommandParser will be working.
 - * **adistributor** - The interface with methods which the parser may call.

METHODS

- *getArguments*

```
public String getArguments( )
```

- **Usage**

- * Returns the sentence on which the parser is working

- **Returns** - An array of words on which the parser is working

- *parse*

```
public void parse( )
```

- **Usage**

- * Try to parse the sentence in **arguments**. The method also interprets the sentence and calls the appropriate method in the appropriate interface. When the sentence could not be parsed, an exception is thrown.

- **Exceptions**

- * **exceptions.SyntaxException** - Thrown when the sentence could not be parsed.

CLASS DIOProcessor

This component has the function to process inputs from the outside world and process outputs from the other components of the dispatcher.

DECLARATION

<pre>public class DIOProcessor extends java.lang.Object implements interfaces.IDInputProcessor, interfaces.IDOutputProcessor</pre>

910 **CONSTRUCTORS**

- *DIOProcessor*
public DIOProcessor()

- **Usage**

915 * Constructs a new instance of this class

METHODS

- *begin*
public void begin()

- **Usage**

920 * This method is called right after the component is created and the interfaces
 are set.

-
- *Input*
public void Input(java.lang.String [] args)

- *Output*
public void Output(java.lang.String str)

- 930 • *setInterfaces*
**public void setInterfaces(interfaces.IDCCommandInput
idCCommandInput, interfaces.IDACommandInput idACommandInput)**

- **Usage**

 * Sets all interfaces this component may use.

- **Parameters**

935 * idCCommandInput -
 * idACommandInput -

3.3 Package aIOProcessor

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
AIOProcessor	38
<i>This component manages the input received from and output given to the user.</i>	
CommandArgument	39
<i>...no description...</i>	
CommandErrors	40
<i>Class containing all error messages and descriptions that can be outputted by the CommandLineParser</i>	
CommandParser	42
<i>...no description...</i>	
<hr/>	

3.3.1 Classes

CLASS **AIOProcessor**

This component manages the input received from and output given to the user. It redirects input to and receives output from the Distributor.

DECLARATION

```
public class AIOProcessor
extends java.lang.Object
implements interfaces.IAInputProcessor, interfaces.IAOutputProcessor
```

CONSTRUCTORS

- *AIOProcessor*
public AIOProcessor()
 - **Usage**
 - * Constructs a new instance of this class

METHODS

- *begin*
public void begin()
 - **Usage**
 - * This method is called right after the component is created and the interfaces are set.
- *Input*
public void Input(java.lang.String [] args)
- *Output*
public void Output(java.lang.String str)
- *setInterfaces*
public void setInterfaces(interfaces.IADistributorForIOProcessor arg0)
 - **Usage**
 - * Sets references to all interfaces this component might need
 - **Parameters**
 - * **arg0** -

CLASS `CommandArgument`

985

DECLARATION

```
public class CommandArgument
extends java.lang.Object
```

990

FIELDS

- `public static final String ROLE_APPPROV`
 - Constant for the paramter 'application provider' in a `CommandArgument` of type role.
- 995 • `public static final String ROLE_JOBPROV`
 - Constant for the paramter 'job provider' in a `CommandArgument` of type role.
- `public static final int TYPE_APPLICATION`
 - A type constant which defines a `CommandArgument` as an application argument
- `public static final int TYPE_PROJECT`
 - 1000 – A type constant which defines a `CommandArgument` as a project argument
- `public static final int TYPE_ROLE`
 - A type constant which defines a `CommandArgument` as a role argument
- `public static final int TYPE_TIME`
 - A type constant which defines a `CommandArgument` as a time argument
- 1005 • `public static final int TYPE_USER`
 - A type constant which defines a `CommandArgument` as a user argument

CONSTRUCTORS

- *CommandArgument*
 - 1010 `public CommandArgument(java.lang.String arg)`
 - **Usage**
 - * Constructs a new `CommandArgument` from a given string. It first checks if the string has the right form and if so the new object is constructed
 - **Parameters**
 - 1015 * The - string to be 'put in' the `CommandArgument`

METHODS

• *getParam*

```
public String getParam( )
```

– Usage

* Returns the parameter of the `CommandArgument`.

– Returns - The parameter of the `CommandArgument`• *getType*

```
public int getType( )
```

– Usage

* Returns the type of the `CommandArgument`.

– Returns - The type of the `CommandArgument`• *isValidArgument*

```
public static boolean isValidArgument( java.lang.String arg )
```

– Usage

* Checks if a string has the right form of an argument. The string should be at least 5 characters long and have a ':' as a fourth character.

– Parameters

* `arg` -

– Returns -

• *isValidRole*

```
public boolean isValidRole( )
```

– Usage

* This method checks whether the `CommandArgument` is of type role and if the parameter is a valid role constant.

– Returns - Whether or not the `CommandArgument` contains a valid role.CLASS `CommandErrors`

Class containing all error messages and descriptions that can be outputted by the `CommandLineParser`

DECLARATION

```
public class CommandErrors
extends java.lang.Object
```

FIELDS

- 1055 • public static final String ERR_COMMANDS
 - An error message that should be put when an invalid command was entered.
- public static final String ERR_HELP_ADD_INTERVAL
 - The description belonging to the command "add interval"
- public static final String ERR_HELP_EXIT
 - 1060 – The description belonging to the command "exit"
- public static final String ERR_HELP_HELP
 - The description belonging to the command "help"
- public static final String ERR_HELP_LIST_PROJECTS
 - The description belonging to the command "list projects"
- 1065 • public static final String ERR_HELP_LIST_USING
 - The description belonging to the command "list using"
- public static final String ERR_HELP_REMOVE_INTERVAL
 - The description belonging to the command "remove interval"
- public static final String ERR_INVALID_INTERVAL
 - 1070 –
- public static final String ERR_SYNTAX_ADD_INTERVAL
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_EXIT
 - 1075 – An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_HELP
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- 1080 • public static final String ERR_SYNTAX_LIST_PROJECTS
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_LIST_USING
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- 1085 • public static final String ERR_SYNTAX_REMOVE_INTERVAL
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.

CONSTRUCTORS

- *CommandErrors*
`public CommandErrors()`

METHODS

- *getArgumentExpected*
`public static String getArgumentExpected(java.lang.String syntax)`

- **Usage**

- * Returns an error message indicating that there is at least one argument missing at the end of the line and the syntax of a specified command

- **Parameters**

- * `syntax` - The syntax

- **Returns** - String with the error message

- *getInvalidSyntaxError*
`public static String getInvalidSyntaxError(java.lang.String token, java.lang.String syntax)`

- **Usage**

- * Returns an error message that parsing failed on a specific token.

- **Parameters**

- * `token` - The token onw hich parsing failed
 - * `syntax` - The correct syntax of the command on which parsing failed

- **Returns** - The error message

- *getTooManyArgumentsError*
`public static String getTooManyArgumentsError(java.lang.String syntax)`

- **Usage**

- * Returns an error message indicating that there is at least one argument too much at the end of the line and the syntax of a specified command

- **Parameters**

- * `syntax` - The syntax

- **Returns** - String with the error message

CLASS `CommandParser`

1125 **DECLARATION**

```
public class CommandParser
extends java.lang.Object
```

CONSTRUCTORS

1130

- *CommandParser*

```
public CommandParser( java.lang.String [] words,
interfaces.IADistributorForIOProcessor distributor,
aIOProcessor.AIOProcessor processor )
```

1135

- **Usage**

- * Constructs a new instance of the CommandParser

- **Parameters**

- * **words** - The sentence (words) on which the CommandParser will be working.
 - * **adistributor** - The interface with methods which the parser may call.

1140 **METHODS**

- *getArguments*

```
public String getArguments( )
```

- **Usage**

1145

- * Returns the sentence on which the parser is working

- **Returns** - An array of words on which the parser is working

- *parse*

```
public void parse( )
```

1150

- **Usage**

- * Try to parse the sentence in **arguments**. The method also interprets the sentence and calls the appropriate method in the appropriate interface. When the sentence could not be parsed, an exception is thrown.

- **Exceptions**

1155

- * **exceptions.SyntaxInvalidException** - Thrown when the sentence could not be parsed.

3.4 Package dAgentManager

Package Contents *Page*

Classes

DAgentManager 45

The Agent Manager has several functions: - Receive the commands from the Agent Translator or the IOProcessor - Perform the desired actions - Update/Read the database - Send the requested results back to the Agent Translator or the IOProcessor - Manages the agent/job status

1160

3.4.1 Classes

CLASS **DAgentManager**

1165 The Agent Manager has several functions: - Receive the commands from the Agent Translator or the
IOProcessor - Perform the desired actions - Update/Read the database - Send the requested results
back to the Agent Translator or the IOProcessor - Manages the agent/job status

DECLARATION

```
public class DAgentManager
extends java.lang.Object
implements interfaces.IDACommandInput, interfaces.IDResourceProviderCommands
```

1170

FIELDS

- public static final int JOB_RUNNING
—
- 1175 • public static final int JOB_FINISHED
—
- public static final int JOB_1xFAILED
—
- public static final int JOB_2xFAILED
1180 —
- public static final int JOB_3xFAILED
—
- public static final int JOB_NEW
—
- 1185 • public static final int JOB_1xFAILED_RUNNING
—
- public static final int JOB_2xFAILED_RUNNING
—

CONSTRUCTORS

- *DAgentManager*
`public DAgentManager()`
 - **Usage**
 - * Constructor

METHODS

- *begin*
`public void begin()`
 - **Usage**
 - * Method that is called when an instance has been created and the interfaces have been set

-
- *generateString*
`public String generateString(int length)`
 - **Usage**
 - * Generates a random string, which can be used as a agentID
 - **Parameters**
 - * `length`, - the length of a string
 - **Returns** - a string of random characters [a-zA-Z0-9“-“_]

-
- *listProjects*
`public String listProjects()`

- *Poll*
`public PollResponse Poll(java.lang.String agentID, types.PollRequest prq)`

-
- *setInterfaces*
`public void setInterfaces(interfaces.IDQueryHandler idQueryHandler, interfaces.IDOutputProcessor idOutputProcessor)`

- **Usage**
 - * Sets references to all interfaces this component might need
- **Parameters**
 - * `idQueryHandler` -
 - * `idOutputProcessor` -

-
- *Shutdown*
`public void Shutdown()`
 - **Usage**

- 1230 * IDACCommandInput methods
-
- *updateFailed*
 public void updateFailed()
 - **Usage**
 - 1235 * This method can be called when it's time for a timeout
 - **Exceptions**
 - * `java.sql.SQLException` -

3.5 Package aDistributor

Package Contents *Page*

Classes

ADistributor 49

*This component receives tasks from the IOProcessor and the JobScheduler
and passes it to the Translator.*

1240

3.5.1 Classes

CLASS **ADistributor**

1245 This component receives tasks from the IOProcessor and the JobScheduler and passes it to the Translator. This component is separated from the Translator for multithreading reasons.

DECLARATION

```
public class ADistributor
extends java.lang.Object
implements interfaces.IADistributorForIOProcessor,
interfaces.IADistributorForJobScheduler, java.lang.Runnable
```

1250

FIELDS

- public static final String JOB_FINISHED
 - Constant for polling
- 1255 • public static final String JOB_FAILED
 - Constant for polling
- public static final int JOBREQUESTMODE
 - constant for job mode
- public static final int JOBFINISHEDMODE
 - 1260 – constant for job mode
- public static final int JOBFAILEDMODE
 - constant for job mode
- public static final int JOBCALCULATING
 - constant for job mode

1265 CONSTRUCTORS

- *ADistributor*
public **ADistributor**()
 - **Usage**
 - 1270 * Creates a new instance of the CCommunicator object

METHODS

• *AddInterval*

```
public void AddInterval( types.Interval interval )
```

• *begin*

```
public void begin( )
```

– Usage

* Method that is called when an instance has been created and the interfaces have been set

• *RemoveInterval*

```
public void RemoveInterval( types.Interval interval )
```

• *run*

```
public void run( )
```

• *sendListProjects*

```
public void sendListProjects( )
```

• *sendPoll*

```
public void sendPoll( )
```

• *sendPoll*

```
public void sendPoll( int jobmode )
```

• *setInterfaces*

```
public void setInterfaces( interfaces.IAResourceProviderCommands  
iaTranslator, interfaces.IAJobSchedulerForDistributor iaJobScheduler,  
interfaces.IAOutputProcessor iaOutputProcessor )
```

– Usage

* Sets references to all interfaces this component might need

– Parameters

* iaTranslator -
* iaJobScheduler -
* iaOutputProcessor -

• *StartScheduler*

```
public void StartScheduler( )
```

• *StopScheduler*

```
public void StopScheduler( )
```

3.6 Package commands

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
AgentCommands	52
...no description...	
ClientCommands	52
...no description...	
<hr/>	

1315

3.6.1 Classes

CLASS **AgentCommands**

DECLARATION

```
public class AgentCommands
extends java.lang.Object
```

FIELDS

- public static final String POLL
 - Constant representing an agent command
- public static final String LIST_PROJECTS
 - Constant representing an agent command

CONSTRUCTORS

- *AgentCommands*
public **AgentCommands**()

CLASS **ClientCommands**

DECLARATION

```
public class ClientCommands
extends java.lang.Object
```

FIELDS

- public static final String SET_PASSWORD
 - Constant representing a Client Command for every user
- public static final String ADD_USER

- Constant representing a Client Command for every user
- public static final String LIST_USERS
 - Constant representing a Client Command for every user
- 1350 • public static final String INFO_USER
 - Constant representing a Client Command for every user
- public static final String INFO_PROJECT
 - Constant representing a Client Command for every user
- public static final String INFO_JOB
 - 1355 – Constant representing a Client Command for every user
- public static final String INFO_DATASET
 - Constant representing a Client Command for every user
- public static final String INFO_APPLICATION
 - Constant representing a Client Command for every user
- 1360 • public static final String APPROVE_JOB_PROVIDER_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- public static final String DISAPPROVE_JOB_PROVIDER_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- public static final String APPROVE_DATA_PROVIDER
 - 1365 – Constant representing a Client Command for a project admin
- public static final String DISAPPROVE_DATA_PROVIDER
 - Constant representing a Client Command for a project admin
- public static final String APPROVE_APPLICATION_PROVIDER
 - Constant representing a Client Command for a project admin
- 1370 • public static final String DISAPPROVE_APPLICATION_PROVIDER
 - Constant representing a Client Command for a project admin
- public static final String ALLOW_OWN_APPLICATION
 - Constant representing a Client Command for a project admin
- public static final String DISALLOW_OWN_APPLICATION
 - 1375 – Constant representing a Client Command for a project admin
- public static final String ALLOW_OWN_DATA
 - Constant representing a Client Command for a project admin

- public static final String DISALLOW_OWN_DATA
 - Constant representing a Client Command for a project admin
- 1380 • public static final String REMOVE_JOB_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- public static final String LIST_JOB_PROVIDERS_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- 1385 • public static final String LIST_APPLICATION_PROVIDERS_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- public static final String LIST_DATA_PROVIDERS_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- public static final String LIST_APPLICATIONS_FOR_PROJECT_ADMIN
 - 1390 – Constant representing a Client Command for a project admin
- public static final String LIST_DATASETS_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- public static final String LIST_PROJECTS_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- 1395 • public static final String LIST_JOBS_FOR_PROJECT_ADMIN
 - Constant representing a Client Command for a project admin
- public static final String REMOVE_USER_FOR_SYSTEM_ADMIN
 - Constant representing a Client Command for a system admin
- public static final String ADD_APPLICATION_PROVIDER
 - 1400 – Constant representing a Client Command for a system admin
- public static final String REMOVE_APPLICATION_PROVIDER
 - Constant representing a Client Command for a system admin
- public static final String ADD_DATA_PROVIDER
 - Constant representing a Client Command for a system admin
- 1405 • public static final String REMOVE_DATA_PROVIDER
 - Constant representing a Client Command for a system admin
- public static final String ADD_PROJECT_ADMIN
 - Constant representing a Client Command for a system admin
- public static final String REMOVE_PROJECT_ADMIN

- 1410 – Constant representing a Client Command for a system admin
- public static final String ADD_PROJECT
 - Constant representing a Client Command for a system admin
- public static final String REMOVE_PROJECT
 - Constant representing a Client Command for a system admin
- 1415 • public static final String LIST_JOB_PROVIDERS_FOR_SYSTEM_ADMIN
 - Constant representing a Client Command for a system admin
- public static final String LIST_APPLICATION_PROVIDERS_FOR_SYSTEM_ADMIN
 - Constant representing a Client Command for a system admin
- public static final String LIST_DATA_PROVIDERS_FOR_SYSTEM_ADMIN
 - 1420 – Constant representing a Client Command for a system admin
- public static final String LIST_PROJECT_ADMINS_FOR_SYSTEM_ADMIN
 - Constant representing a Client Command for a system admin
- public static final String LIST_APPLICATIONS_FOR_SYSTEM_ADMIN
 - Constant representing a Client Command for a system admin
- 1425 • public static final String LIST_DATASETS_FOR_SYSTEM_ADMIN
 - Constant representing a Client Command for a system admin
- public static final String LIST_PROJECTS_FOR_SYSTEM_ADMIN
 - Constant representing a Client Command for a system admin
- public static final String LIST_JOBS_FOR_SYSTEM_ADMIN
 - 1430 – Constant representing a Client Command for a system admin
- public static final String OFFER_JOB
 - Constant representing a Client Command for a job provider
- public static final String REMOVE_JOB_FOR_JOB_PROVIDER
 - Constant representing a Client Command for a job provider
- 1435 • public static final String LIST_APPLICATIONS_FOR_JOB_PROVIDER
 - Constant representing a Client Command for a job provider
- public static final String LIST_DATASETS_FOR_JOB_PROVIDER
 - Constant representing a Client Command for a job provider
- public static final String LIST_PROJECTS_FOR_JOB_PROVIDER
 - 1440 – Constant representing a Client Command for a job provider

- public static final String LIST_JOBS_FOR_JOB_PROVIDER
 - Constant representing a Client Command for a job provider
- public static final String ADD_APPLICATION
 - Constant representing a Client Command for an application provider
- 1445 • public static final String REMOVE_APPLICATION
 - Constant representing a Client Command for an application provider
- public static final String APPROVE_APPLICATION_TO_PROJECT
 - Constant representing a Client Command for an application provider
- public static final String DISAPPROVE_APPLICATION_TO_PROJECT
 - 1450 – Constant representing a Client Command for an application provider
- public static final String LIST_APPLICATIONS_FOR_APPLICATION_PROVIDER
 - Constant representing a Client Command for an application provider
- public static final String LIST_PROJECTS_FOR_APPLICATION_PROVIDER
 - Constant representing a Client Command for an application provider
- 1455 • public static final String
LIST_JOBS_THAT_USE_APP_FOR_APPLICATION_PROVIDER
 - Constant representing a Client Command for an application provider
- public static final String ADD_DATA
 - Constant representing a Client Command for a data provider
- 1460 • public static final String REMOVE_DATA
 - Constant representing a Client Command for a data provider
- public static final String APPROVE_DATA_TO_PROJECT
 - Constant representing a Client Command for a data provider
- public static final String DISAPPROVE_DATA_TO_PROJECT
 - 1465 – Constant representing a Client Command for a data provider
- public static final String LIST_DATASETS_FOR_DATA_PROVIDER
 - Constant representing a Client Command for a data provider
- public static final String LIST_PROJECTS_FOR_DATA_PROVIDER
 - Constant representing a Client Command for a data provider
- 1470 • public static final String LIST_JOBS_THAT_USE_DATA_FOR_DATA_PROVIDER
 - Constant representing a Client Command for a data provider

CONSTRUCTORS

- *ClientCommands*

1475

```
public ClientCommands( )
```

3.7 Package interfaces

Package Contents

Page

Interfaces

IACommunicator 61

The methods in this interface should implement a way to send GSPs.

IADataManager 61

The methods in this interface should allow other component to ask the implementing component to download and or upload files from/to a certain location.

IADistributorForIOProcessor 62

This interface is specifically meant for the IOProcessor-component.

IADistributorForJobScheduler 63

This interface is specifically meant for the JobScheduler-component.

IAInputProcessor 64

*There is only one method in this interface: **Input**.*

IAJobExecutor 64

The methods in this interface should allow components to tell the implementing component to execute a Job on the machine on which the Agent is running.

IAJobSchedulerForDataManager 65

This interface is specifically meant for the DataManager-component.

IAJobSchedulerForDistributor 66

This interface is specifically meant for the Distributor-component.

IAJobSchedulerForJobExecutor 67

This interface is specifically meant for the JobExecutor-component.

IAOutputProcessor 68

*The method **Output** can be used to ask the implementing component to print a sequence of characters to standard output.*

IAResourceProviderCommands 68

The methods in this interface correspond to the commands a Resource Provider can send to the dispatcher.

ICApplicationProviderCommands 69

The methods in this interface correspond to the commands an Application Provider can send to the dispatcher.

ICCommunicator 71

The methods in this interface should implement a way to send GSPs.

ICDataProviderCommands 72

The methods in this interface correspond to the commands a Data Provider can send to the dispatcher.

ICInputProcessor 73

*There is only one method in this interface: **Input**.*

ICJobProviderCommands 74

The methods in this interface correspond to the commands a Job Provider can send to the dispatcher.

	ICOutputProcessor	75
	<i>The method <code>Output</code> can be used to ask the implementing component to print a sequence of characters to standard output.</i>	
1510	ICProjectAdminCommands	75
	<i>The methods in this interface correspond to the commands a Project Admin can send to the dispatcher.</i>	
	ICSystemAdminCommands	80
	<i>The methods in this interface correspond to the commands a System Admin can send to the dispatcher.</i>	
1515	ICUserCommands	84
	<i>The methods in this interface correspond to the commands a User can send to the dispatcher.</i>	
	IDACommandInput	85
	<i>This interface handles commands (mostly generated by a UI).</i>	
	IDApplicationProviderCommands	86
1520	<i>The methods in this interface correspond to the commands an Application Provider can send (using the Client) to the dispatcher.</i>	
	IDCCommandInput	89
	<i>This interface handles commands (mostly generated by a UI).</i>	
	IDDataProviderCommands	90
	<i>The methods in this interface correspond to the commands a Data Provider can send (using the Client) to the dispatcher.</i>	
1525	IDInputProcessor	93
	<i>There is only one method in this interface: <code>Input</code>.</i>	
	IDJobProviderCommands	93
	<i>The methods in this interface correspond to the commands a Job Provider can send (using the Client) to the dispatcher.</i>	
	IDOutputProcessor	96
1530	<i>The method <code>Output</code> can be used to ask the implementing component to print a sequence of characters to standard output.</i>	
	IDPackageInput	97
	<i>Use the <code>PerformCommand</code> method to transmit data.</i>	
	IDProjectAdminCommands	97
	<i>The methods in this interface correspond to the commands a Project Admin can send (using the Client) to the dispatcher.</i>	
1535	IDQueryHandler	106
	<i>The methods in this interface deal with database queries.</i>	
	IDResourceProviderCommands	107
	<i>The methods in this interface correspond to the commands a Resource Provider can send (using the Agent) to the dispatcher.</i>	
	IDSystemAdminCommands	108
1540	<i>The methods in this interface correspond to the commands a System Admin can send (using the Client) to the dispatcher.</i>	
	IDUserCommands	116
	<i>The methods in this interface correspond to the commands a User can send (using the Client) to the dispatcher.</i>	

3.7.1 Interfaces

INTERFACE **IACommunicator**

The methods in this interface should implement a way to send GSPs.

DECLARATION

```
public interface IACommunicator
```

METHODS

- *send*

```
public GRP send( genericpackages.GSP  pack, java.net.InetAddress  addr  
                )
```

- **Usage**

- * This method should be called when a GSP package (a list of pairs of variables and values) needs to be send to another host. To accomplish this, the method uses converts the GSPs into messages that conform to the protocol and then sends those to the given host. The method then blocks and waits for a response. If a response is not received in time, an exception is thrown. After the response is received the data is converted to a GRP (combination of a string containing the status code and a string containing a message) and returned.

- **Parameters**

- * **sendpackage** - The GSP to send
 - * **addr** - The destination

- **Returns** - A GRP with the results

INTERFACE **IADDataManager**

The methods in this interface should allow other component to ask the implementing component to download and or upload files from/to a certain location.

DECLARATION

```
public interface IADDataManager
```

METHODS

- *download*

```
public DownOrUploadStatus download( java.lang.String from,  
java.lang.String to )
```

- **Usage**

- * This method should be called when data located in from should be downloaded to to.

- **Parameters**

- * **from** - Location to download from
 - * **to** - Location to download to

- *upload*

```
public DownOrUploadStatus upload( java.lang.String from,  
java.lang.String to )
```

- **Usage**

- * This method should be called when data located in from should be uploaded to to.

- **Parameters**

- * **from** - Location to upload from
 - * **to** - Location to upload to

INTERFACE IADistributorForIOProcessor

This interface is specifically meant for the IOProcessor-component. Using the methods in this interface an IOProcessor can ask a Distributor-component to forward its requests.

DECLARATION

```
public interface IADistributorForIOProcessor
```

METHODS

- *AddInterval*

```
public void AddInterval( types.Interval interval )
```

- **Usage**

- * This method calls AddInterval(Interval a, Interval b) in IAJobScheduler which is implemented in JobScheduler.

- **Parameters**

1615 * **a** - vraagteken
 * **b** - vraagteken

• *RemoveInterval*
 public void RemoveInterval(types.Interval interval)

1620 – **Usage**
 * This method calls RemoveInterval(Interval a, Interval b) in IAJobScheduler
 which is implemented in JobScheduler.

 – **Parameters**
 * **a** - vraagteken
1625 * **b** - vraagteken

• *sendListProjects*
 public void sendListProjects()

 – **Usage**
1630 * This methods asks the dispatcher for a list of projects

• *StartScheduler*
 public void StartScheduler()

 – **Usage**
1635 * This method calls Start() in IAJobScheduler which is implemented in
 JobScheduler.

• *StopScheduler*
 public void StopScheduler()

1640 – **Usage**
 * This method calls Stop() in IAJobScheduler which is implemented in
 JobScheduler.

INTERFACE **IADistributorForJobScheduler**

1645 This interface is specifically meant for the JobScheduler-component. Using the methods in this interface a JobScheduler can ask a Distributor-component to forward its requests.

DECLARATION

public interface IADistributorForJobScheduler
--

1650

METHODS

- *sendPoll*
`public void sendPoll()`

1655 – **Usage**

- * This method calls `sendPoll()` in `IATranslator` which is implemented in `Translator`.

INTERFACE **IAInputProcessor**

1660 There is only one method in this interface: **Input**. It processes input.

DECLARATION

<code>public interface IAInputProcessor</code>
--

1665

METHODS

- *Input*
`public void Input(java.lang.String [] args)`

1670 – **Usage**

- * This method should be called when the outside world has a sequence of characters ready to process. This sequence will be stored in the `args` variable.

– **Parameters**

- * `args` - The sequences of characters ready to process.

INTERFACE **IAJobExecutor**

1675 The methods in this interface should allow components to tell the implementing component to execute a **Job** on the machine on which the Agent is running.

DECLARATION

1680

<code>public interface IAJobExecutor</code>

METHODS

- *startJob*

```
public int startJob( java.lang.String path, java.lang.String []  
myproglis )
```

- **Usage**

- * This method should be called when a job needs to start executing.

- **Returns** - int A unique identifier

- *stopJob*

```
public void stopJob( int job )
```

- **Usage**

- * This method should be called when a job needs to stop executing.

- **Parameters**

- * job - The job identifier

INTERFACE IAJobSchedulerForDataManager

This interface is specifically meant for the DataManager-component. Using the methods in this interface a DataManager can tell a JobScheduler-component that it has finished the tasks that it was requested to perform.

DECLARATION

```
public interface IAJobSchedulerForDataManager
```

METHODS

- *giveDownloadResult*

```
public void giveDownloadResult( int status )
```

- **Usage**

- * This method should be called when the status of the download order has changed.

- **Parameters**

- * status - The new status of the download status

- *giveUploadResult*

```
public void giveUploadResult( int status )
```

– **Usage**

- * This method should be called when the status of the upload order has changed.

– **Parameters**

- * **status** - The new status of the upload status

INTERFACE **IAJobSchedulerForDistributor**

This interface is specifically meant for the Distributor-component. Using the methods in this interface a Distributor can tell a JobScheduler-component that it has finished the tasks that it was requested to perform.

DECLARATION

```
public interface IAJobSchedulerForDistributor
```

METHODS

- *AddInterval*

```
public void AddInterval( types.Interval interval )
```

– **Usage**

- * This method is called when an interval should be added when the resource can be used.

– **Parameters**

- * **a** -
- * **b** -

- *getCurrentmode*

```
public int getCurrentmode( )
```

– **Usage**

- * This method is called when the current mode of the running job is needed

– **Returns** - int

- *receivePoll*

```
public void receivePoll( JSDL.JSDL job )
```

– **Usage**

- * This method is called when the answer to a poll is starting a job.

– **Parameters**

- * **Object** -

- *RemoveInterval*

```
public void RemoveInterval( types.Interval  interval )
```

- **Usage**

- * This method is called when an interval should be removed when the resource can be used.

- **Parameters**

- * **a** -
 - * **b** -

- *StartScheduling*

```
public void StartScheduling( )
```

- **Usage**

- * This method is called when the JobScheduler should begin with scheduling jobs.

- *StopJob*

```
public void StopJob( )
```

- **Usage**

- * This method is called when the answer to a poll is cancelling the job.

- *StopScheduling*

```
public void StopScheduling( )
```

- **Usage**

- * This method is called when the JobScheduler should stop with scheduling jobs.

INTERFACE **IAJobSchedulerForJobExecutor**

This interface is specifically meant for the JobExecutor-component. Using the methods in this interface a JobExecutor can tell a JobScheduler-component that it has finished the tasks that it was requested to perform.

DECLARATION

public interface IAJobSchedulerForJobExecutor

METHODS

- *giveResult*

`public void giveResult(int status)`

- **Usage**

* This method should be called when the status of a job has changed.

- **Parameters**

* **status** - The new status of the job

INTERFACE **IAOutputProcessor**

The method `Output` can be used to ask the implementing component to print a sequence of characters to standard output.

DECLARATION

<code>public interface IAOutputProcessor</code>

METHODS

- *Output*

`public void Output(java.lang.String str)`

- **Usage**

* This method should be called when another component wants to send a sequence of characters to standard output. This sequence will be printed in the console.

- **Parameters**

* **str** - The **String** to send to standard output

INTERFACE **IAResourceProviderCommands**

The methods in this interface correspond to the commands a Resource Provider can send to the dispatcher.

DECLARATION

<code>public interface IAResourceProviderCommands</code>
--

METHODS

- *sendListProjects*

```
public String sendListProjects( )
```

- **Usage**

- * This method should be called when the resource wants to see a list of projects currently available in the system.

- **Exceptions**

- * `exceptions.CommunicationException` - Thrown when something went wrong

- *sendPoll*

```
public PollResponse sendPoll( types.PollRequest polling )
```

- **Usage**

- * This method should be called when a resource polls the dispatcher for work, tells the dispatcher that he is busy or tells the dispatcher that the job has been completed.

- **Parameters**

- * `AgentID` - the AgentID of the Agent
 - * `polling` - The PollRequest

- **Returns** - According to the PollRequest, this will be an URL of a job or a message.

- **Exceptions**

- * `exceptions.CommunicationException` - Thrown when something went wrong
 - * -

INTERFACE `ICApplicationProviderCommands`

The methods in this interface correspond to the commands an Application Provider can send to the dispatcher.

DECLARATION

```
public interface ICApplicationProviderCommands
```

METHODS

- *sendAddApplication*

```
public void sendAddApplication( java.lang.String file,
    java.lang.String name, java.lang.String description )
```

– **Usage**

- * This method should be called to add an application to the SPINGRID system.

– **Parameters**

- * **application** - The URL to the application
- * **appname** - The name of the application

• *sendApproveApplicationToProject*

public void sendApproveApplicationToProject(java.lang.String application, java.lang.String projectname)

– **Usage**

- * This method should be called to approve an application for the given project.

– **Parameters**

- * **application** - The identification of the application
- * **projectname** - The name of the project

• *sendDisapproveApplicationToProject*

public void sendDisapproveApplicationToProject(java.lang.String application, java.lang.String projectname)

– **Usage**

- * This method should be called to disapprove an application for the given project.

– **Parameters**

- * **application** - The identification of the application
- * **projectname** - The name of the project

• *sendListApplicationsForApplicationProvider*

public void sendListApplicationsForApplicationProvider()

– **Usage**

- * This method should be called when a list of applications is requested by an application provider

• *sendListJobsForApplicationProvider*

public void sendListJobsForApplicationProvider()

– **Usage**

- * This method should be called when a list of jobs is requested by an application provider

• *sendListJobsForApplicationProvider*

public void sendListJobsForApplicationProvider(java.lang.String project)

– **Usage**

- 1905 * This method should be called when a list of jobs is requested by an
 application provider in a specific project
- **Parameters**
- * **project** - The name of the project to use as restriction for the list
- 1910 • *sendListProjectsForApplicationProvider*
 public void **sendListProjectsForApplicationProvider**()
- **Usage**
- * This method should be called when a list of projects is requested by an
 application provider
- 1915 • *sendRemoveApplication*
 public void **sendRemoveApplication**(java.lang.String **application**)
- **Usage**
- * This method should be called to remove an application from the SPINGRID
1920 system.
- **Parameters**
- * **application** - The identification of the application

INTERFACE **ICCommunicator**

1925 The methods in this interface should implement a way to send GSPs.

DECLARATION

public interface ICCommunicator
--

1930 METHODS

- *send*
 public GRP **send**(genericpackages.GSP **pack**, java.net.InetAddress **addr**
)
- 1935 – **Usage**
- * This method should be called when a GSP package (a list of pairs of
 variables and values) needs to be send to another host. To accomplish this,
 the method uses converts the GSPs into messages that conform to the
 protocol and then sends those to the given host. The method then blocks
1940 and waits for a response. If a response is not received in time, an exception is
 thrown. After the response is received the data is converted to a GRP
 (combination of a string containing the status code and a string containing a
 message) and returned.

– **Parameters**

- * **sendpackage** - The GSP to send
- * **addr** - The destination

– **Returns** - A GRP with the results

INTERFACE **ICDataProviderCommands**

The methods in this interface correspond to the commands a Data Provider can send to the dispatcher.

DECLARATION

```
public interface ICDataProviderCommands
```

METHODS

• *sendAddData*

```
public void sendAddData( java.net.URL [] data, java.lang.String
name, java.lang.String description )
```

– **Usage**

- * This method should be called to add data to the SPINGRID system.

– **Parameters**

- * **data** - The URL of the data
- * **dataname** - The name of the data

• *sendApproveDataToProject*

```
public void sendApproveDataToProject( java.lang.String data,
java.lang.String projectname )
```

– **Usage**

- * This method should be called to approve data for the given project.

– **Parameters**

- * **data** - The identification of the data
- * **projectname** - The name of the project

• *sendDisapproveDataToProject*

```
public void sendDisapproveDataToProject( java.lang.String data,
java.lang.String projectname )
```

– **Usage**

- * This method should be called to disapprove data for the given project.

– **Parameters**

- * **data** - The identification of the data
- * **projectname** - The name of the project

• *sendListDataSetsForDataProvider*

public void sendListDataSetsForDataProvider()

– **Usage**

- * This method should be called when a data provider requests a list of his datasets

• *sendListProjectsForDataProvider*

public void sendListProjectsForDataProvider()

– **Usage**

- * This method should be called when a data provider wants to see a list of projects he trusts his datasets to

• *sendRemoveData*

public void sendRemoveData(java.lang.String data)

– **Usage**

- * This method should be called to remove data from the SPINGRID system.

– **Parameters**

- * **data** - The identification of the data

INTERFACE **ICInputProcessor**

There is only one method in this interface: **Input**. It processes input.

DECLARATION

public interface ICInputProcessor

METHODS

• *Input*

public void Input(java.lang.String [] args)

– **Usage**

- * This method should be called when the outside world has a sequence of characters ready to process. This sequence will be stored in the args variable.

– **Parameters**

- * **args** - The sequences of characters ready to process.

INTERFACE **ICJobProviderCommands**

The methods in this interface correspond to the commands a Job Provider can send to the dispatcher.

DECLARATION

```
public interface ICJobProviderCommands
```

METHODS

- *sendListApplicationsForJobProvider*

```
public void sendListApplicationsForJobProvider( )
```

- *sendListApplicationsForJobProvider*

```
public void sendListApplicationsForJobProvider( java.lang.String  
project )
```

- **Parameters**

- * **project** - The name of the project to use as restriction for the list

- *sendListDataSetsForJobProvider*

```
public void sendListDataSetsForJobProvider( )
```

- *sendListDataSetsForJobProvider*

```
public void sendListDataSetsForJobProvider( java.lang.String project  
)
```

- **Parameters**

- * **project** - The name of the project to use as restriction for the list

- *sendListJobsForJobProvider*

```
public void sendListJobsForJobProvider( )
```

- *sendListJobsForJobProvider*

```
public void sendListJobsForJobProvider( java.lang.String project )
```

- **Parameters**

- * **project** - The name of the project to use as restriction for the list

- *sendListProjectsForJobProvider*

```
public void sendListProjectsForJobProvider( )
```

- *sendOfferJob*

```
public void sendOfferJob( java.lang.String projectname,  
java.lang.String job )
```

– **Usage**

* This method should be called to offer a job for the given project.

– **Parameters**

* **projectname** - The name of the project

* **job** - The job to offer

• *sendRemoveJobForJobProvider*

public void sendRemoveJobForJobProvider(java.lang.String job)

– **Usage**

* This method should be called to remove a job from the SPINGRID system.

– **Parameters**

* **job** - The identification of the job

INTERFACE **ICOutputProcessor**

The method **Output** can be used to ask the implementing component to print a sequence of characters to standard output.

DECLARATION

```
public interface ICOutputProcessor
```

METHODS

• *Output*

public void Output(java.lang.String str)

– **Usage**

* This method should be called when another component wants to send a sequence of characters to standard output. This sequence will be printed in the console.

– **Parameters**

* **str** - The **String** to send to standard output

INTERFACE **ICProjectAdminCommands**

The methods in this interface correspond to the commands a Project Admin can send to the dispatcher.

DECLARATION

2095

```
public interface ICProjectAdminCommands
```

METHODS

2100

- *sendAllowOwnData*

```
public void sendAllowOwnData( java.lang.String  projectname,
                             java.lang.String  jobprovider )
```

- **Usage**

- * Command to allow a jobprovider to send his own data

2105

- **Parameters**

- * **projectname** -
 - * **jobprovider** -

- *sendApproveApplication*

2110

```
public void sendApproveApplication( java.lang.String  projectname,
                                    java.lang.String  application )
```

- **Usage**

- * This method should be called to approve an application for the project.

- **Parameters**

2115

- * **projectname** - The name of the project
 - * **application** - The name of the application

- *sendApproveApplicationProvider*

2120

```
public void sendApproveApplicationProvider( java.lang.String
                                             projectname, java.lang.String  applicationprovider )
```

- **Usage**

- * This method should be called to approve an application provider for the project.

- **Parameters**

2125

- * **projectname** - The name of the project
 - * **applicationprovider** - The name of the application provider

- *sendApproveDataProvider*

2130

```
public void sendApproveDataProvider( java.lang.String  projectname,
                                     java.lang.String  dataprovider )
```

- **Usage**

- * This method should be called to approve a data provider for the project.

- **Parameters**

- * **projectname** - The name of the project

2135 * **dataprovider** - The name of the dataprovider

• *sendApproveJobProvider*
 public void **sendApproveJobProvider**(java.lang.String **username**,
 java.lang.String **projectname**)

2140 – **Usage**
 * This method should be called to approve a job provider for the project.

 – **Parameters**
 * **username** - The name of the user
 * **projectname** - The name of the project to approve a user for as JobProvider

2145 • *sendDisallowOwnData*
 public void **sendDisallowOwnData**(java.lang.String **projectname**,
 java.lang.String **jobprovider**)

 – **Usage**
 * Command to disallow a jobprovider to send his own data

2150 – **Parameters**
 * **projectname** -
 * **jobprovider** -

2155 • *sendDisapproveApplication*
 public void **sendDisapproveApplication**(java.lang.String **projectname**,
 java.lang.String **application**)

 – **Usage**
 * This method should be called to disapprove an application for the project.

2160 – **Parameters**
 * **projectname** - The name of the project
 * **application** - The name of the application

2165 • *sendDisapproveApplicationProvider*
 public void **sendDisapproveApplicationProvider**(java.lang.String
projectname, java.lang.String **applicationprovider**)

 – **Usage**
 * This method should be called to disapprove an application provider for the
 project.

2170 – **Parameters**
 * **projectname** - The name of the project
 * **applicationprovider** - The name of the application provider

2175 • *sendDisapproveDataProvider*
 public void **sendDisapproveDataProvider**(java.lang.String
projectname, java.lang.String **dataprovider**)

 – **Usage**

* This method should be called to disapprove a data provider for the project.

– **Parameters**

* **projectname** - The name of the project
 * **dataprovder** - The name of the dataprovder

• *sendDisapproveJobProvider*

public void **sendDisapproveJobProvider**(java.lang.String username,
 java.lang.String projectname)

– **Usage**

* This method should be called to disapprove a job provider for the project.

– **Parameters**

* **username** - The username of the JobProvider
 * **projectname** - The name of the project

• *sendListApplicationProvidersForProjectAdmin*

public void **sendListApplicationProvidersForProjectAdmin**()

– **Usage**

* This method should be called to request a list of application providers for the project admin

• *sendListApplicationsForProjectAdmin*

public void **sendListApplicationsForProjectAdmin**()

– **Usage**

* This method should be called to request a list of applications for the project admin

• *sendListApplicationsForProjectAdmin*

public void **sendListApplicationsForProjectAdmin**(java.lang.String project)

– **Usage**

* This method should be called to request a list of applications for the project admin

– **Parameters**

* **project** - The name of the project to use as restriction for the list

• *sendListDataProvidersForProjectAdmin*

public void **sendListDataProvidersForProjectAdmin**()

– **Usage**

* This method should be called to request a list of data providers for the project admin

• *sendListDataSetsForProjectAdmin*

public void **sendListDataSetsForProjectAdmin**()

– Usage

- * This method should be called to request a list of datasets for the project admin

• *sendListDataSetsForProjectAdmin*

public void **sendListDataSetsForProjectAdmin**(java.lang.String project)

– Usage

- * This method should be called to request a list of datasets for the project admin for a specific project

– Parameters

- * **project** - The name of the project to use as restriction for the list

• *sendListJobProvidersForProjectAdmin*

public void **sendListJobProvidersForProjectAdmin**()

– Usage

- * This method should be called to request a list of job providers for the project admin

• *sendListJobProvidersForProjectAdmin*

public void **sendListJobProvidersForProjectAdmin**(java.lang.String project)

– Parameters

- * **project** - The name of the project to use as restriction for the list

• *sendListJobsForProjectAdmin*

public void **sendListJobsForProjectAdmin**()

– Usage

- * This method should be called to request a list of jobs for the project

• *sendListJobsForProjectAdmin*

public void **sendListJobsForProjectAdmin**(java.lang.String project)

– Parameters

- * **project** - The name of the project to use as restriction for the list

• *sendListProjectsForProjectAdmin*

public void **sendListProjectsForProjectAdmin**()

– Usage

- * This method should be called when a list of projects is requested for a project admin

• *sendRemoveJobForProjectAdmin*

public void **sendRemoveJobForProjectAdmin**(java.lang.String job)

– **Usage**

2265 * This method should be called to remove a job from the SPINGRID system.

– **Parameters**

* **job** - The identification of the job

INTERFACE **ICSystemAdminCommands**

2270 The methods in this interface correspond to the commands a System Admin can send to the dispatcher.

DECLARATION

2275 `public interface ICSystemAdminCommands`

METHODS

- *sendAddApplicationProvider*

`public void sendAddApplicationProvider(java.lang.String username)`

2280 – **Usage**

* This method should be called to add an application provider to the SPINGRID system.

– **Parameters**

* **username** - The name of the user that should be given the role appprovider

- *sendAddDataProvider*

`public void sendAddDataProvider(java.lang.String username)`

2290 – **Usage**

* This method should be called to add a data provider to the SPINGRID system.

– **Parameters**

* **username** - The name of the user that should be given the role dataprovider

- *sendAddProject*

2295 `public void sendAddProject(java.lang.String projectname,
java.lang.String username)`

– **Usage**

* This method should be called to add a project to the SPINGRID system.
The given user is made project admin of the project.

2300 – **Parameters**

- * **projectname** - A name for the project
- * **username** - The user to make initial project admin

- *sendAddProjectAdmin*

2305 **public void sendAddProjectAdmin(java.lang.String username,**
java.lang.String projectname)

- **Usage**

- * This method should be called to add a project admin to the SPINGRID system.

- **Parameters**

- * **username** - The name of the user to make project admin
- * **projectname** - The name of the project

- *sendListApplicationProvidersForSystemAdmin*

2315 **public void sendListApplicationProvidersForSystemAdmin()**

- **Usage**

- * This method should be called to request a list of application providers for the systemadmin

- *sendListApplicationsForSystemAdmin*

2320 **public void sendListApplicationsForSystemAdmin()**

- **Usage**

- * This method should be called to request a list of applications for the systemadmin

- *sendListApplicationsForSystemAdmin*

2325 **public void sendListApplicationsForSystemAdmin(java.lang.String**
project)

- **Usage**

- * This method should be called to request a list of applications for the system admin in a specific project

- **Parameters**

- * **project** - The name of the project to use as restriction for the list

- *sendListDataProvidersForSystemAdmin*

2335 **public void sendListDataProvidersForSystemAdmin()**

- **Usage**

- * This method should be called to request a list of data providers for the systemadmin

- *sendListDataSetsForSystemAdmin*

2340 **public void sendListDataSetsForSystemAdmin()**

- **Usage**

2345 * This method should be called to request a list of datasets for the system
 admin

• *sendListDataSetsForSystemAdmin*
 public void **sendListDataSetsForSystemAdmin**(java.lang.String project)

2350 – **Usage**
 * This method should be called to request a list of datasets for the system
 admin in a specific project

 – **Parameters**
 * **project** - The name of the project to use as restriction for the list

2355 • *sendListJobProvidersForSystemAdmin*
 public void **sendListJobProvidersForSystemAdmin**()

 – **Usage**
 * This method should be called to request a list of job providers for the
 2360 systemadmin

• *sendListJobProvidersForSystemAdmin*
 public void **sendListJobProvidersForSystemAdmin**(java.lang.String
 project)

2365 – **Usage**
 * This method should be called to request a list of job providers for the
 systemadmin in a specific project

 – **Parameters**
 * **project** - The name of the project to use as restriction for the list

2370 • *sendListJobsForSystemAdmin*
 public void **sendListJobsForSystemAdmin**()

 – **Usage**
 * This method should be called to request a list all jobs in the system for the
 2375 system admin

• *sendListJobsForSystemAdmin*
 public void **sendListJobsForSystemAdmin**(java.lang.String project)

 – **Parameters**
 * **project** - The name of the project to use as restriction for the list

2380 • *sendListProjectAdminsForSystemAdmin*
 public void **sendListProjectAdminsForSystemAdmin**()

 – **Usage**
 2385 * This method should be called to request a list of project admins for the
 systemadmin

- *sendListProjectsForSystemAdmin*

```
public void sendListProjectsForSystemAdmin( )
```

2390 – **Usage**

- * This method should be called to request a list all projects in the system for the system admin

- *sendRemoveApplicationProvider*

2395

```
public void sendRemoveApplicationProvider( java.lang.String  
username )
```

 – **Usage**

- * This method should be called to remove an application provider from the SPINGRID system.

2400 – **Parameters**

- * **username** - The name of the user of who the role appprovider should be removed from

- *sendRemoveDataProvider*

2405

```
public void sendRemoveDataProvider( java.lang.String  username )
```

 – **Usage**

- * This method should be called to remove a data provider from the SPINGRID system.

 – **Parameters**

- 2410 * **username** - The name of the user of who the role dataprovider should be removed from

- *sendRemoveProject*

```
public void sendRemoveProject( java.lang.String  projectname )
```

2415 – **Usage**

- * This method should be called to remove a project from the SPINGRID system. If there are job providers who where only allowed to provide jobs for the given project then they will loose the role of job provider. If the project admin was only project admin of the given project then he will loose the role of project admin.
- 2420

 – **Parameters**

- * **projectname** - The name of the project to be removed

- *sendRemoveProjectAdmin*

2425

```
public void sendRemoveProjectAdmin( java.lang.String  username,  
java.lang.String  projectname )
```

 – **Usage**

- * This method should be called to remove a project admin from the SPINGRID system.

2430 – **Parameters**

- * **username** - The name of the user to unmake project admin
- * **projectname** - The name of the project

- *sendRemoveUserForSystemAdmin*

2435 public void **sendRemoveUserForSystemAdmin**(java.lang.String
 username)

 – **Usage**

- * This method should be called to remove a user from the system.

 – **Parameters**

- 2440 * **username** - The name of the user

INTERFACE **ICUserCommands**

The methods in this interface correspond to the commands a User can send to the dispatcher.

DECLARATION

2445 public interface **ICUserCommands**

METHODS

2450 • *sendAddUser*

 public void **sendAddUser**(java.lang.String **username**, java.lang.String
 password)

 – **Usage**

- * This method should be called to add a new user to the system.

2455 – **Parameters**

- * **username** - The name of the user
- * **password** - The password of the new user

- *sendInfoApplication*

2460 public void **sendInfoApplication**(java.lang.String **applicationname**)

 – **Usage**

- * Sends a request for information about an application

 – **Parameters**

- * **applicationname** -

- *sendInfoDataSet*

2465 public void **sendInfoDataSet**(java.lang.String **datasetname**)

- **Usage**
 - * Sends a request for information about a dataset

2470

- **Parameters**
 - * `datasetname` -
-

- *sendInfoJob*

```
public void sendInfoJob( java.lang.String  jobnumber )
```

2475

- **Usage**
 - * Sends a request for information about a job
 - **Parameters**
 - * `jobnumber` -
-

2480

- *sendInfoProject*

```
public void sendInfoProject( java.lang.String  projectname )
```

- **Usage**
 - * Sends a request for information about a project
 - **Parameters**
 - * `projectname` -
-

2485

- *sendInfoUser*

```
public void sendInfoUser( java.lang.String  username )
```

- **Usage**
 - * Sends a request for information about a project
 - **Parameters**
 - * `username` -
-

2490

- *sendListUsers*

```
public void sendListUsers( )
```

- **Usage**
 - * Sends the command to list users
-

2495

- *sendSetPassword*

```
public void sendSetPassword( java.lang.String  password )
```

- **Usage**
 - * This method should be called when a user wants to change his password.
- **Parameters**
 - * `password` - The password in which to change

2500

2505 INTERFACE **IDACCommandInput**

This interface handles commands (mostly generated by a UI). For now, only the **Shutdown** method exists.

DECLARATION

2510

```
public interface IDACCommandInput
```

METHODS

2515

- *Shutdown*

```
public void Shutdown( )
```

- **Usage**

- * Signals the ClientManager that the dispatcher will shut down.

INTERFACE IDApplicationProviderCommands

2520

The methods in this interface correspond to the commands an Application Provider can send (using the Client) to the dispatcher.

DECLARATION

2525

```
public interface IDApplicationProviderCommands
```

METHODS

- *AddApplication*

2530

```
public void AddApplication( java.lang.String user, java.lang.String
hash, java.lang.String applicationxml, java.lang.String
applicationname, java.lang.String description )
```

- **Usage**

- * This method should be called to add an application to the SPINGRID system.

2535

- **Parameters**

- * **application** - The URL with the location of the application.
 - * **applicationname** - The name of the application.

- **Exceptions**

2540

- * **java.sql.SQLException** - Connection to the database failed
 - * **exceptions.UserException** - The user does not exist
 - * **exceptions.RoleException** - The user does not have the role applicationprovider

2545 * `exceptions.SyntaxInvalidException` - The XML does not conform to the
 correct syntax
 * `exceptions.ApplicationException` - Application with that name already
 exists
 * `exceptions.SPINGRIDException` - UTF-8 encoding is not supported

2550 • *ApproveApplicationToProject*
 public void **ApproveApplicationToProject**(java.lang.String user,
 java.lang.String hash, java.lang.String application, java.lang.String
 projectname)

 – Usage
2555 * This method should be called to approve an application for the given project.

 – Parameters
 * `application` - The identification of the application.
 * `projectname` - The name of the project.

 – Exceptions
2560 * `java.sql.SQLException` - Connection to the database failed
 * `exceptions.UserException` - The user does not exist
 * `exceptions.ProjectException` - The project does not exist
 * `exceptions.RoleException` - User does not have the role
 applicationprovider
2565 * `exceptions.ApplicationException` - The application does not exist
 * `exceptions.TrustException` - The appprovider is not trusted by any
 projectadmin in the project

2570 • *DisapproveApplicationToProject*
 public void **DisapproveApplicationToProject**(java.lang.String user,
 java.lang.String hash, java.lang.String application, java.lang.String
 projectname)

 – Usage
2575 * This method should be called to disapprove an application for the given
 project.

 – Parameters
 * `application` - The identification of the application.
 * `projectname` - The name of the project.

 – Exceptions
2580 * `java.sql.SQLException` - Connection to the database failed
 * `exceptions.UserException` - User does not exist
 * `exceptions.ProjectException` - Project does not exist
 * `exceptions.ApplicationException` - Application does not exist
 * `exceptions.TrustException` - Trust does not exist

2585 • *listApplicationsForApplicationProvider*
 public String **listApplicationsForApplicationProvider**(java.lang.String
 user, java.lang.String hash)

 – Parameters

2590 * **user** - The user that sent the command.
 * **hash** - The authentication hash.
 – **Returns** - A %-seperated list
 – **Exceptions**
 * **exceptions.AuthenticationException** - The hash-parameter does not
 2595 match the hash of the username, password and command
 * **exceptions.UserException** - The user does not exist
 * **java.sql.SQLException** - Connection to the database failed
 * **exceptions.RoleException** - The user does not have the role application
 provider

2600

• *listJobsThatUseAppForApplicationProvider*

**public String listJobsThatUseAppForApplicationProvider(
 java.lang.String user, java.lang.String hash)**

2605

– **Parameters**
 * **user** - The user that sent the command.
 * **hash** - The authentication hash.
 – **Returns** - A %-seperated list
 – **Exceptions**
 * **exceptions.AuthenticationException** - The hash-parameter does not
 2610 match the hash of the username, password and command
 * **exceptions.UserException** - The user does not exist
 * **java.sql.SQLException** - Connection to the database failed
 * **exceptions.RoleException** - The user does not have the role application
 provider

2615

• *listJobsThatUseAppForApplicationProvider*

**public String listJobsThatUseAppForApplicationProvider(
 java.lang.String user, java.lang.String hash, java.lang.String
 project)**

2620

– **Parameters**
 * **user** - The user that sent the command.
 * **hash** - The authentication hash.
 * **project** - The name of the project to use as restriction for the list
 – **Returns** - A %-seperated list
 – **Exceptions**
 * **exceptions.AuthenticationException** - The hash-parameter does not
 2625 match the hash of the username, password and command
 * **exceptions.UserException** - The user does not exist
 * **java.sql.SQLException** - Connection to the database failed
 2630 * **exceptions.RoleException** - The user does not have the role application
 provider
 * **exceptions.ProjectException** - The project does not exist

2635

• *listProjectsForApplicationProvider*

**public String listProjectsForApplicationProvider(java.lang.String
 user, java.lang.String hash)**

- **Parameters**

- * **user** - The user that sent the command.
- * **hash** - The authentication hash.

- **Returns** - A %-seperated list

- **Exceptions**

- * **exceptions.AuthenticationException** - The hash-parameter does not match the hash of the username, password and command
- * **exceptions.UserException** - The user does not exist
- * **java.sql.SQLException** - Connection to the database failed
- * **exceptions.RoleException** - The user does not have the role application provider

- *RemoveApplication*

```
public void RemoveApplication( java.lang.String user,  
java.lang.String hash, java.lang.String application )
```

- **Usage**

- * This method should be called to remove an application from the SPINGRID system.

- **Parameters**

- * **application** - The identification of the application.

- **Exceptions**

- * **java.sql.SQLException** - A database error ocurred
- * **exceptions.UserException** - User does not exist
- * **exceptions.ApplicationException** - Application does not exist
- * **exceptions.RoleException** - User is not an application provider

INTERFACE **IDCCCommandInput**

This interface handles commands (mostly generated by a UI). For now, only the **Shutdown** method exists.

DECLARATION

```
public interface IDCCCommandInput
```

METHODS

- *Shutdown*

```
public void Shutdown( )
```

- **Usage**

- * Signals the ClientManager that the dispatcher will shut down.

INTERFACE **IDDataProviderCommands**

The methods in this interface correspond to the commands a Data Provider can send (using the Client) to the dispatcher.

DECLARATION

```
public interface IDDataProviderCommands
```

METHODS

- *AddData*

```
public void AddData( java.lang.String user, java.lang.String hash,
java.net.URL [] data, java.lang.String dataname, java.lang.String
description )
```

- **Usage**

- * This method should be called to add data to the SPINGRID system.

- **Parameters**

- * **data** - The data to be added
 - * **dataname** - The name of the application

- **Exceptions**

- * **java.sql.SQLException** - Connection to the database failed
 - * **exceptions.UserException** - User does not exist.
 - * **exceptions.RoleException** - User does not have the dataprovider role.
 - * **exceptions.DataSetException** -

- *ApproveDataToProject*

```
public void ApproveDataToProject( java.lang.String user,
java.lang.String hash, java.lang.String data, java.lang.String
projectname )
```

- **Usage**

- * This method should be called to approve data for the given project.

- **Parameters**

- * **data** - The data to be approved
 - * **projectname** - The name of the project

- **Exceptions**

- * **exceptions.ProjectException** - The project does not exist
 - * **exceptions.RoleException** - The user is not a data provider
 - * **exceptions.DataSetException** - The dataset does not exist
 - * **java.sql.SQLException** - Connection to the database failed
 - * **exceptions.TrustException** - Already trust.

* `exceptions.UserException` - User does not exist.

- *DisapproveDataToProject*

public void **DisapproveDataToProject**(java.lang.String user,
java.lang.String hash, java.lang.String data, java.lang.String
projectname)

- **Usage**

- * This method should be called to disapprove data for the given project.

- **Parameters**

- * `data` - The data to be disapproved
 - * `projectname` - The name of the project

- **Exceptions**

- * `java.sql.SQLException` - Connection to the database failed
 - * `exceptions.UserException` - User does not exist.
 - * `exceptions.ProjectException` - Project does not exist.
 - * `exceptions.RoleException` - User does not have role `dataprov`
 - * `exceptions.DataSetException` - `DataSet` does not exist or it is not owned by User
 - * `exceptions.TrustException` - `DataSet` did not trust Project

- *listDataSetsForDataProvider*

public String **listDataSetsForDataProvider**(java.lang.String user,
java.lang.String hash)

- **Parameters**

- * `user` - The user that sent the command.
 - * `hash` - The authentication hash.

- **Returns** - A %-seperated list

- **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
 - * `exceptions.UserException` - The user does not exist
 - * `java.sql.SQLException` - Connection to the database failed
 - * `exceptions.RoleException` - The user does not have the role data provider

- *listJobsThatUseDataForDataProvider*

public String **listJobsThatUseDataForDataProvider**(java.lang.String
user, java.lang.String hash)

- **Parameters**

- * `user` - The user that sent the command.
 - * `hash` - The authentication hash.

- **Returns** - A %-seperated list

- **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
 - * `exceptions.UserException` - The user does not exist

- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role data provider

- *listJobsThatUseDataForDataProvider*

`public String listJobsThatUseDataForDataProvider(java.lang.String user, java.lang.String hash, java.lang.String project)`

- **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.
- * `project` - The name of the project to use as restriction for the list

- **Returns** - A %-seperated list

- **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role data provider
- * `exceptions.ProjectException` - The project does not exist

- *listProjectsForDataProvider*

`public String listProjectsForDataProvider(java.lang.String user, java.lang.String hash)`

- **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.

- **Returns** - A %-seperated list

- **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role data provider

- *RemoveData*

`public void RemoveData(java.lang.String user, java.lang.String hash, java.lang.String data)`

- **Usage**

- * This method should be called to remove data from the SPINGRID system.

- **Parameters**

- * `data` - The data to be removed

- **Exceptions**

- * `java.sql.SQLException` - A database error occurred
- * `exceptions.UserException` - User does not exist
- * `exceptions.DataSetException` - dataset does not exist
- * `exceptions.RoleException` - User is not a dataset provider

INTERFACE **IDInputProcessor**

There is only one method in this interface: **Input**. It processes input.

DECLARATION

2810

```
public interface IDInputProcessor
```

METHODS

2815

- *Input*

```
public void Input( java.lang.String [] args )
```

- **Usage**

- * This method should be called when the outside world has a sequence of characters ready to process. This sequence will be stored in the args variable.

2820

- **Parameters**

- * **args** - The sequences of characters ready to process.

INTERFACE **IDJobProviderCommands**

2825 The methods in this interface correspond to the commands a Job Provider can send (using the Client) to the dispatcher.

DECLARATION

```
public interface IDJobProviderCommands
```

2830

METHODS

- *listApplicationsForJobProvider*

```
public String listApplicationsForJobProvider( java.lang.String user,
java.lang.String hash )
```

2835

- **Parameters**

- * **user** - The user that sent the command.

- * **hash** - The authentication hash.

- **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role job provider

• *listApplicationsForJobProvider*

```
public String listApplicationsForJobProvider( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

– **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.
- * `project` - The name of the project to use as restriction for the list

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role job provider
- * `exceptions.ProjectException` - The project does not exist

• *listDataSetsForJobProvider*

```
public String listDataSetsForJobProvider( java.lang.String user,
java.lang.String hash )
```

– **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role job provider

• *listDataSetsForJobProvider*

```
public String listDataSetsForJobProvider( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

– **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.
- * `project` - The name of the project to use as restriction for the list

– **Returns** - A %-seperated list

– **Exceptions**

- 2885 * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role job provider
- 2890 * `exceptions.ProjectException` - The project does not exist

• *listJobsForJobProvider*

```
public String listJobsForJobProvider( java.lang.String user,
java.lang.String hash )
```

– **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

- 2900 * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role job provider

• *listJobsForJobProvider*

```
public String listJobsForJobProvider( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

– **Parameters**

- 2910 * `user` - The user that sent the command.
- * `hash` - The authentication hash.
- * `project` - The name of the project to use as restriction for the list

– **Returns** - A %-seperated list

– **Exceptions**

- 2915 * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role job provider
- 2920 * `exceptions.ProjectException` - The project does not exist

• *listProjectsForJobProvider*

```
public String listProjectsForJobProvider( java.lang.String user,
java.lang.String hash )
```

– **Parameters**

- 2925 * `user` - The user that sent the command.
- * `hash` - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role job provider

• *OfferJob*

```
public void OfferJob( java.lang.String user, java.lang.String hash,
java.lang.String projectname, java.lang.String job )
```

– **Usage**

- * This method should be called to offer a job for the given project.

– **Parameters**

- * `projectname` - The name of the project.
- * `job` - The URL of the job.

– **Exceptions**

- * `exceptions.ProjectException` - Project does not exist
- * `exceptions.UserException` - User does not exist
- * `exceptions.RoleException` - User does not have the role JobProvider
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.JobException` - Lets kapot
- * `exceptions.SPINGRIDException` - UTF-8 encoding

• *RemoveJobForJobProvider*

```
public void RemoveJobForJobProvider( java.lang.String user,
java.lang.String hash, java.lang.String job )
```

– **Usage**

- * This method should be called to remove a job from the SPINGRID system.

– **Parameters**

- * `job` - The identification of the job to be removed.

– **Exceptions**

- * `SQLExceptionRoleException` - A database error occurred
- * `exceptions.UserException` - User does not exist
- * `exceptions.RoleException` - User is not a job provider
- * `exceptions.JobException` - Job does not exist
- * `exceptions.ProjectException` - Project of the job does not exist

INTERFACE **IDOutputProcessor**

The method `Output` can be used to ask the implementing component to print a sequence of characters to standard output.

DECLARATION

```
public interface IDOutputProcessor
```

METHODS

- *Output*

```
public void Output( java.lang.String str )
```

- **Usage**

* This method should be called when another component wants to send a sequence of characters to standard output. This sequence will be printed in the console.

- **Parameters**

* **str** - The **String** to send to standard output

INTERFACE IDPackageInput

Use the **PerformCommand** method to transmit data.

DECLARATION

```
public interface IDPackageInput
```

METHODS

- *PerformCommand*

```
public GRP PerformCommand( genericpackages.GSP Package )
```

- **Usage**

* Returns the result package of a command, after the package is translated into a command and it is performed

- **Parameters**

* **Package** - A GSP with data to transmit

- **Returns** - A GRP with the results of the transmission

INTERFACE IDProjectAdminCommands

The methods in this interface correspond to the commands a Project Admin can send (using the Client) to the dispatcher.

DECLARATION

```
public interface IDProjectAdminCommands
```

METHODS

- *AllowOwnApplication*

```
public void AllowOwnApplication( java.lang.String user,
    java.lang.String hash, java.lang.String projectname,
    java.lang.String jobprovider )
```

- **Usage**

- * This method should be called to allow a job provider to use his private applications for the project.

- **Parameters**

- * **projectname** - The name of the project.
 - * **jobprovider** - The username of the jobprovider.

- **Exceptions**

- * **exceptions.AuthenticationException** - When the specified hash is incorrect.
 - * **exceptions.RoleException** - When the specified user does not have the role of ProjectAdmin in the specified project.
 - * **exceptions.TrustException** - When the specified user already trusts the specified target user.
 - * **exceptions.UserException** - When the specified user or target user does not exist.
 - * **exceptions.ProjectException** - When the specified project does not exist.
 - * **java.sql.SQLException** - The database is not reachable

- *AllowOwnData*

```
public void AllowOwnData( java.lang.String user, java.lang.String
    hash, java.lang.String projectname, java.lang.String jobprovider )
```

- **Usage**

- * This method should be called to allow a job provider to use his private applications for the project.

- **Parameters**

- * **projectname** - The name of the project.
 - * **jobprovider** - The username of the jobprovider.

- **Exceptions**

- * **exceptions.AuthenticationException** - When the specified hash is incorrect.

3045 * `exceptions.RoleException` - When the specified user does not have the
 role of `ProjectAdmin` in the specified project.
 * `exceptions.TrustException` - When the specified user already trusts the
 specified target user.
 * `exceptions.UserException` - When the specified user or target user does
 not exist.
3050 * `exceptions.ProjectException` - When the specified project does not exist.
 * `java.sql.SQLException` - The database is not reachable

- *ApproveApplicationProvider*

3055 public void **ApproveApplicationProvider**(java.lang.String user,
 java.lang.String hash, java.lang.String projectname,
 java.lang.String applicationprovider)

- Usage

- * This method should be called to approve an application provider for the
project.

- Parameters

- * `projectname` - The name of the project.
 - * `applicationprovider` - The username of the applicationprovider.

- Exceptions

3065 * `exceptions.AuthenticationException` - When the specified hash is
 incorrect.
 * `exceptions.ProjectException` - When the specified project does not exist.
 * `exceptions.RoleException` - When the specified user does not have the role
 `ProjectAdmin` or the target user does not have the role `ApplicationProvider`.
 * `exceptions.UserException` - When the specified user or target user does
3070 not exist.
 * `exceptions.TrustException` - When the specified user already trusts the
 specified target user.
 * `java.sql.SQLException` - The database is not reachable

- *ApproveDataProvider*

3075 public void **ApproveDataProvider**(java.lang.String user,
 java.lang.String hash, java.lang.String projectname,
 java.lang.String datapvider)

- Usage

3080 * This method should be called to approve a data provider for the project.

- Parameters

- * `projectname` - The name of the project.
 - * `datapvider` - The username of the datapvider.

- Exceptions

3085 * `exceptions.AuthenticationException` - When the specified hash is
 incorrect.
 * `exceptions.ProjectException` - When the specified project does not exist.
 * `exceptions.RoleException` - When the specified user does not have the
 role `ProjectAdmin` or the target user does not have the role `DataProvider`.

3090 * `exceptions.UserException` - When the specified user or target user does
 not exist.
 * `exceptions.TrustException` - When the specified user already trusts the
 specified target user.
 * `java.sql.SQLException` - The database is not reachable

3095

- *ApproveJobProviderForProjectAdmin*
 public void **ApproveJobProviderForProjectAdmin**(java.lang.String
 user, java.lang.String hash, java.lang.String projectname,
 java.lang.String jobprovider)

3100 – **Usage**
 * This method should be called to approve a job provider for the project.

 – **Parameters**
 * `projectname` - The name of the project.
 * `jobprovider` - The username of the jobprovider.

3105 – **Exceptions**
 * `exceptions.AuthenticationException` - When the specified hash is
 incorrect.
 * `exceptions.RoleException` - When the specified user does not have the
 role of ProjectAdmin in the specified project.
 3110 * `exceptions.TrustException` - When the specified user already trusts the
 specified target user.
 * `exceptions.UserException` - When the specified user or target user does
 not exist.
 * `exceptions.ProjectException` - When the specified project does not exist.
 3115 * `java.sql.SQLException` - The database is not reachable

- *DisallowOwnApplication*
 public void **DisallowOwnApplication**(java.lang.String user,
 java.lang.String hash, java.lang.String projectname,
 java.lang.String jobprovider)

3120 – **Usage**
 * This method should be called to disallow a job provider to use his private
 applications for the project.

 – **Parameters**
 3125 * `projectname` - The name of the project.
 * `jobprovider` - The username of the jobprovider.

 – **Exceptions**
 * `exceptions.AuthenticationException` - When the specified hash is
 incorrect.
 3130 * `exceptions.RoleException` - When the specified user does not have the
 role of ProjectAdmin in the specified project or the target user does not have
 the role PrivateApplicationProvider.
 * `exceptions.TrustException` - When the specified user did not trust the
 specified target user.

- 3135 * `exceptions.UserException` - When the specified user or target user does not exist.
- * `exceptions.ProjectException` - When the specified project does not exist.
- * `java.sql.SQLException` - The database is not reachable

3140 • *DisallowOwnData*

`public void DisallowOwnData(java.lang.String user, java.lang.String hash, java.lang.String projectname, java.lang.String jobprovider)`

– **Usage**

- 3145 * This method should be called to disallow a job provider to use his private applications for the project.

– **Parameters**

- * `projectname` - The name of the project.
- * `jobprovider` - The username of the jobprovider.

– **Exceptions**

- 3150 * `exceptions.AuthenticationException` - When the specified hash is incorrect.
- * `exceptions.RoleException` - When the specified user does not have the role of ProjectAdmin in the specified project or the target user does not have the role PrivateApplicationProvider.
- 3155 * `exceptions.TrustException` - When the specified user did not trust the specified target user.
- * `exceptions.UserException` - When the specified user or target user does not exist.
- * `exceptions.ProjectException` - When the specified project does not exist.
- 3160 * `java.sql.SQLException` - The database is not reachable

• *DisapproveApplicationProvider*

`public void DisapproveApplicationProvider(java.lang.String user, java.lang.String hash, java.lang.String projectname, java.lang.String applicationprovider)`

– **Usage**

- * This method should be called to disapprove an application provider for the project.

– **Parameters**

- 3170 * `projectname` - The name of the project.
- * `applicationprovider` - The username of the applicationprovider.

– **Exceptions**

- * `exceptions.AuthenticationException` - When the specified hash is incorrect.
- 3175 * `exceptions.RoleException` - When the specified user does not have the role of ProjectAdmin in the specified project or the target user does not have the role ApplicationProvider.
- * `exceptions.TrustException` - When the specified user did not trust the specified target user.

- * `exceptions.UserException` - When the specified user or target user does not exist.
- * `exceptions.ProjectException` - When the specified project does not exist.
- * `java.sql.SQLException` - The database is not reachable

3185 • *DisapproveDataProvider*

```
public void DisapproveDataProvider( java.lang.String user,
java.lang.String hash, java.lang.String projectname,
java.lang.String dataprovider )
```

– Usage

- * This method should be called to disapprove a data provider for the project.

– Parameters

- * `projectname` - The name of the project.
- * `dataprovider` - The username of the dataprovider.

– Exceptions

- 3195 * `exceptions.AuthenticationException` - When the specified hash is incorrect.
- * `exceptions.RoleException` - When the specified user does not have the role of ProjectAdmin in the specified project or the target user does not have the role DataProvider.
- 3200 * `exceptions.TrustException` - When the specified user did not trust the specified target user.
- * `exceptions.UserException` - When the specified user or target user does not exist.
- * `exceptions.ProjectException` - When the specified project does not exist.
- 3205 * `java.sql.SQLException` - The database is not reachable

• *DisapproveJobProviderForProjectAdmin*

```
public void DisapproveJobProviderForProjectAdmin( java.lang.String
user, java.lang.String hash, java.lang.String projectname,
3210 java.lang.String jobprovider )
```

– Usage

- * This method should be called to disapprove a job provider for the project.

– Parameters

- 3215 * `projectname` - The name of the project.
- * `jobprovider` - The username of the jobprovider.

– Exceptions

- * `exceptions.AuthenticationException` - When the specified hash is incorrect.
- 3220 * `exceptions.RoleException` - When the specified user does not have the role of ProjectAdmin in the specified project or the target user does not have the role JobProvider.
- * `exceptions.TrustException` - When the specified user did not trust the specified target user.
- 3225 * `exceptions.UserException` - When the specified user or target user does not exist.

- * `exceptions.ProjectException` - When the specified project does not exist.
- * `java.sql.SQLException` - The database is not reachable

- *listApplicationProvidersForProjectAdmin*

3230 `public String listApplicationProvidersForProjectAdmin(java.lang.String`
`user, java.lang.String hash)`

- **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.

3235 – **Returns** - A %-seperated list

- **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- 3240 * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role project admin

- *listApplicationsForProjectAdmin*

3245 `public String listApplicationsForProjectAdmin(java.lang.String user,`
`java.lang.String hash)`

- **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.

- **Returns** - A %-seperated list

3250 – **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- 3255 * `exceptions.RoleException` - The user does not have the role project admin

- *listApplicationsForProjectAdmin*

`public String listApplicationsForProjectAdmin(java.lang.String user,`
`java.lang.String hash, java.lang.String project)`

3260 – **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.
- * `project` - The name of the project to use as restriction for the list

- **Returns** - A %-seperated list

3265 – **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- 3270 * `exceptions.RoleException` - The user does not have the role project admin

* `exceptions.ProjectException` - The project does not exist

- *listDataProvidersForProjectAdmin*

```
public String listDataProvidersForProjectAdmin( java.lang.String
user, java.lang.String hash )
```

– **Parameters**

* `user` - The user that sent the command.
* `hash` - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

* `exceptions.AuthenticationException` - The hash-parameter does not
match the hash of the username, password and command
* `exceptions.UserException` - The user does not exist
* `java.sql.SQLException` - Connection to the database failed
* `exceptions.RoleException` - The user does not have the role project admin

- *listDataSetsForProjectAdmin*

```
public String listDataSetsForProjectAdmin( java.lang.String user,
java.lang.String hash )
```

– **Parameters**

* `user` - The user that sent the command.
* `hash` - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

* `exceptions.AuthenticationException` - The hash-parameter does not
match the hash of the username, password and command
* `exceptions.UserException` - The user does not exist
* `java.sql.SQLException` - Connection to the database failed
* `exceptions.RoleException` - The user does not have the role project admin

- *listDataSetsForProjectAdmin*

```
public String listDataSetsForProjectAdmin( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

– **Parameters**

* `user` - The user that sent the command.
* `hash` - The authentication hash.
* `project` - The name of the project to use as restriction for the list

– **Returns** - A %-seperated list

– **Exceptions**

* `exceptions.AuthenticationException` - The hash-parameter does not
match the hash of the username, password and command
* `exceptions.UserException` - The user does not exist
* `java.sql.SQLException` - Connection to the database failed
* `exceptions.RoleException` - The user does not have the role project admin
* `exceptions.ProjectException` - The project does not exist

- *listJobProvidersForProjectAdmin*

```
public String listJobProvidersForProjectAdmin( java.lang.String user,
java.lang.String hash )
```

- **Parameters**

- * **user** - The user that sent the command.
 - * **hash** - The authentication hash.

- **Returns** - A %-seperated list

- *listJobProvidersForProjectAdmin*

```
public String listJobProvidersForProjectAdmin( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

- **Parameters**

- * **user** - The user that sent the command.
 - * **hash** - The authentication hash.
 - * **project** - The name of the project to use as restriction for the list

- **Returns** - A %-seperated list

- **Exceptions**

- * **exceptions.AuthenticationException** - The hash-parameter does not match the hash of the username, password and command
 - * **exceptions.UserException** - The user does not exist
 - * **java.sql.SQLException** - Connection to the database failed
 - * **exceptions.RoleException** - The user does not have the role project admin
 - * **exceptions.ProjectException** - The project does not exist
-

- *listJobsForProjectAdmin*

```
public String listJobsForProjectAdmin( java.lang.String user,
java.lang.String hash )
```

- **Parameters**

- * **user** - The user that sent the command.
 - * **hash** - The authentication hash.

- **Returns** - A %-seperated list

- **Exceptions**

- * **exceptions.AuthenticationException** - The hash-parameter does not match the hash of the username, password and command
 - * **exceptions.UserException** - The user does not exist
 - * **java.sql.SQLException** - Connection to the database failed
 - * **exceptions.RoleException** - The user does not have the role project admin
-

- *listJobsForProjectAdmin*

```
public String listJobsForProjectAdmin( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

- **Parameters**

- * **user** - The user that sent the command.
 - * **hash** - The authentication hash.
 - * **project** - The name of the project to use as restriction for the list

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role project admin
- * `exceptions.ProjectException` - The project does not exist

• *listProjectsForProjectAdmin*

```
public String listProjectsForProjectAdmin( java.lang.String user,
java.lang.String hash )
```

– **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role project admin

• *RemoveJob*

```
public void RemoveJob( java.lang.String user, java.lang.String
hash, java.lang.String job )
```

– **Usage**

- * This method should be called to remove a job from the SPINGRID system.

– **Parameters**

- * `job` - The identification number of the job.

– **Exceptions**

- * `exceptions.AuthenticationException` - When the specifed hash is incorrect.
- * `java.sql.SQLException` - A database error occured
- * `exceptions.UserException` - User does not exist
- * `exceptions.RoleException` - User is not a project admin
- * `exceptions.JobException` - Job does not exist

INTERFACE IDQueryHandler

The methods in this interface deal with database queries.

DECLARATION

`public interface IDQueryHandler`

3405

METHODS

• *PerformQuery*`public ResultSet PerformQuery(java.lang.String query)`

3410

– **Usage**

* This method should be called when a query (that doesn't change the database) needs to be performed on the database.

– **Parameters**

* `query` - A `String` containing the SQL query to be performed

3415

– **Returns** - A `QueryResult` containing the result of the query• *PerformUpdate*`public int PerformUpdate(java.lang.String query)`

3420

– **Usage**

* This method should be called when an update (that changes the database) needs to be performed on the database.

– **Parameters**

* `query` - A `String` containing the SQL query to be performed

– **Returns** - An `int` containing the number of updated records in the database3425 **INTERFACE IDResourceProviderCommands**

The methods in this interface correspond to the commands a Resource Provider can send (using the Agent) to the dispatcher.

DECLARATION

3430

`public interface IDResourceProviderCommands`

METHODS

3435

• *listProjects*`public String listProjects()`

– **Usage**

- * This method should be called when a resource polls the dispatcher for work, tells the dispatcher that he is busy or tells the dispatcher that the job has been completed.

– **Parameters**

- * **polling** - A PollRequest with the status of the agent

– **Returns** - According to the PollRequest, this will be an URL of a job or a message.

– **Exceptions**

- * `java.sql.SQLException` - A database error occurred

• *Poll*

```
public PollResponse Poll( java.lang.String  AgentID, types.PollRequest
polling )
```

– **Usage**

- * This method should be called when a resource polls the dispatcher for work, tells the dispatcher that he is busy or tells the dispatcher that the job has been completed.

– **Parameters**

- * **polling** - A PollRequest with the status of the agent

– **Returns** - According to the PollRequest, this will be an URL of a job or a message.

INTERFACE IDSystemAdminCommands

The methods in this interface correspond to the commands a System Admin can send (using the Client) to the dispatcher.

DECLARATION

```
public interface IDSystemAdminCommands
```

METHODS

• *AddApplicationProvider*

```
public void AddApplicationProvider( java.lang.String  user,
java.lang.String  hash, java.lang.String  username )
```

– **Usage**

- * This method should be called to add an application provider to the SPINGRID system.

– **Parameters**

- * **username** - The user that will be added as application provider

– **Exceptions**

- * **exceptions.TrustException** - The SystemAdmin already trusts the target user.
- * **exceptions.UserException** - The user or target user does not exist.
- * **exceptions.RoleException** - The user does not have the role SystemAdmin or the target user already has the role AppProv.
- * **java.sql.SQLException** - The database is not reachable

• *AddDataProvider*

```
public void AddDataProvider( java.lang.String user, java.lang.String
hash, java.lang.String username )
```

– **Usage**

- * This method should be called to add a data provider to the SPINGRID system.

– **Parameters**

- * **username** - The user that will be added as data provider

– **Exceptions**

- * **exceptions.TrustException** - The SystemAdmin already trusts the target user.
- * **exceptions.UserException** - The user or target user does not exist.
- * **exceptions.RoleException** - The user does not have the role SystemAdmin or the target user already has the role DataProv.
- * **java.sql.SQLException** - The database is not reachable

• *AddProject*

```
public void AddProject( java.lang.String user, java.lang.String
hash, java.lang.String projectname, java.lang.String username )
```

– **Usage**

- * This method should be called to add a project to the SPINGRID system. The given user is made project admin of the project.

– **Parameters**

- * **projectname** - The name of the project that will be added
- * **username** - The user that will be project admin of the project

– **Exceptions**

- * **exceptions.UserException** - The user or the target user does not exist.
- * **exceptions.ProjectException** - The projectname is already in use.
- * **exceptions.RoleException** - The specified user is not a SystemAdmin.
- * **java.sql.SQLException** - The database is not reachable

• *AddProjectAdmin*

```
public void AddProjectAdmin( java.lang.String user, java.lang.String
hash, java.lang.String username, java.lang.String projectname )
```

– **Usage**

3520 * This method should be called to add a project admin to the SPINGRID
 system.

– **Parameters**

 * **username** - The user that will be added as project admin
 * **projectname** - The project that the project admin will be added to

3525 – **Exceptions**

 * **exceptions.TrustException** - The SystemAdmin already trusts the target
 user.
 * **exceptions.UserException** - The user or target user does not exist.
 * **exceptions.RoleException** - The user does not have the role SystemAdmin
 3530 or the target user already has the role ProjectAdmin in the specified Project.
 * **exceptions.ProjectException** - The specified project does not exist.
 * **java.sql.SQLException** - The database is not reachable

• *listApplicationProvidersForSystemAdmin*

3535 public String listApplicationProvidersForSystemAdmin(java.lang.String
 user, java.lang.String hash)

– **Parameters**

 * **user** - The user that sent the command.
 * **hash** - The authentication hash.

3540 – **Returns** - A %-seperated list

– **Exceptions**

 * **exceptions.AuthenticationException** - The hash-parameter does not
 match the hash of the username, password and command
 * **exceptions.UserException** - The user does not exist
 3545 * **java.sql.SQLException** - Connection to the database failed
 * **exceptions.RoleException** - The user does not have the role system admin

• *listApplicationsForSystemAdmin*

3550 public String listApplicationsForSystemAdmin(java.lang.String user,
 java.lang.String hash)

– **Parameters**

 * **user** - The user that sent the command.
 * **hash** - The authentication hash.

– **Returns** - A %-seperated list

3555 – **Exceptions**

 * **exceptions.AuthenticationException** - The hash-parameter does not
 match the hash of the username, password and command
 * **exceptions.UserException** - The user does not exist
 * **java.sql.SQLException** - Connection to the database failed
 3560 * **exceptions.RoleException** - The user does not have the role system admin

• *listApplicationsForSystemAdmin*

 public String listApplicationsForSystemAdmin(java.lang.String user,
 java.lang.String hash, java.lang.String project)

3565 – **Parameters**

- * **user** - The user that sent the command.
- * **hash** - The authentication hash.
- * **project** - The name of the project to use as restriction for the list
- **Returns** - A %-seperated list
- **Exceptions**
 - * **exceptions.AuthenticationException** - The hash-parameter does not match the hash of the username, password and command
 - * **exceptions.UserException** - The user does not exist
 - * **java.sql.SQLException** - Connection to the database failed
 - * **exceptions.RoleException** - The user does not have the role system admin
 - * **exceptions.ProjectException** - The project does not exist

• *listDataProvidersForSystemAdmin*

```
public String listDataProvidersForSystemAdmin( java.lang.String
user, java.lang.String hash )
```

- **Parameters**
 - * **user** - The user that sent the command.
 - * **hash** - The authentication hash.
- **Returns** - A %-seperated list
- **Exceptions**
 - * **exceptions.AuthenticationException** - The hash-parameter does not match the hash of the username, password and command
 - * **exceptions.UserException** - The user does not exist
 - * **java.sql.SQLException** - Connection to the database failed
 - * **exceptions.RoleException** - The user does not have the role system admin

• *listDataSetsForSystemAdmin*

```
public String listDataSetsForSystemAdmin( java.lang.String user,
java.lang.String hash )
```

- **Parameters**
 - * **user** - The user that sent the command.
 - * **hash** - The authentication hash.
- **Returns** - A %-seperated list
- **Exceptions**
 - * **exceptions.AuthenticationException** - The hash-parameter does not match the hash of the username, password and command
 - * **exceptions.UserException** - The user does not exist
 - * **java.sql.SQLException** - Connection to the database failed
 - * **exceptions.RoleException** - The user does not have the role system admin

• *listDataSetsForSystemAdmin*

```
public String listDataSetsForSystemAdmin( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

- **Parameters**
 - * **user** - The user that sent the command.
 - * **hash** - The authentication hash.

- * **project** - The name of the project to use as restriction for the list
- **Returns** - A %-seperated list
- **Exceptions**
 - * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
 - * `exceptions.UserException` - The user does not exist
 - * `java.sql.SQLException` - Connection to the database failed
 - * `exceptions.RoleException` - The user does not have the role system admin
 - * `exceptions.ProjectException` - The project does not exist

• *listJobProvidersForSystemAdmin*

```
public String listJobProvidersForSystemAdmin( java.lang.String user,
java.lang.String hash )
```

- **Parameters**
 - * **user** - The user that sent the command.
 - * **hash** - The authentication hash.
- **Returns** - A %-seperated list
- **Exceptions**
 - * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
 - * `exceptions.UserException` - The user does not exist
 - * `java.sql.SQLException` - Connection to the database failed
 - * `exceptions.RoleException` - The user does not have the role system admin

• *listJobProvidersForSystemAdmin*

```
public String listJobProvidersForSystemAdmin( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

- **Parameters**
 - * **user** - The user that sent the command.
 - * **hash** - The authentication hash.
 - * **project** - The name of the project to use as restriction for the list
- **Returns** - A %-seperated list
- **Exceptions**
 - * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
 - * `exceptions.UserException` - The user does not exist
 - * `java.sql.SQLException` - Connection to the database failed
 - * `exceptions.RoleException` - The user does not have the role system admin
 - * `exceptions.ProjectException` - The project does not exist

• *listJobsForSystemAdmin*

```
public String listJobsForSystemAdmin( java.lang.String user,
java.lang.String hash )
```

- **Parameters**
 - * **user** - The user that sent the command.
 - * **hash** - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role system admin

• *listJobsForSystemAdmin*

```
public String listJobsForSystemAdmin( java.lang.String user,
java.lang.String hash, java.lang.String project )
```

– **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.
- * `project` - The name of the project to use as restriction for the list

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role system admin
- * `exceptions.ProjectException` - The project does not exist

• *listProjectAdminsForSystemAdmin*

```
public String listProjectAdminsForSystemAdmin( java.lang.String
user, java.lang.String hash )
```

– **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role system admin

• *listProjectsForSystemAdmin*

```
public String listProjectsForSystemAdmin( java.lang.String user,
java.lang.String hash )
```

– **Parameters**

- * `user` - The user that sent the command.
- * `hash` - The authentication hash.

– **Returns** - A %-seperated list

– **Exceptions**

- * `exceptions.AuthenticationException` - The hash-parameter does not match the hash of the username, password and command
- * `exceptions.UserException` - The user does not exist
- * `java.sql.SQLException` - Connection to the database failed
- * `exceptions.RoleException` - The user does not have the role system admin

• *RemoveApplicationProvider*

```
public void RemoveApplicationProvider( java.lang.String user,
java.lang.String hash, java.lang.String username )
```

– **Usage**

- * This method should be called to remove an application provider from the SPINGRID system.

– **Parameters**

- * `username` - The application provider that will be removed

– **Exceptions**

- * `exceptions.TrustException` - The SystemAdmin does not trust the target user.
- * `exceptions.RoleException` - The user does not have the role SystemAdmin or the target user does not have the role AppProv.
- * `exceptions.UserException` - The user or target user does not exist.
- * `java.sql.SQLException` - The database is not reachable

• *RemoveDataProvider*

```
public void RemoveDataProvider( java.lang.String user,
java.lang.String hash, java.lang.String username )
```

– **Usage**

- * This method should be called to remove a data provider from the SPINGRID system.

– **Parameters**

- * `username` - The data provider that will be removed

– **Exceptions**

- * `exceptions.TrustException` - The SystemAdmin does not trust the target user.
- * `exceptions.RoleException` - The user does not have the role SystemAdmin or the target user does not have the role DataProv.
- * `exceptions.UserException` - The user or target user does not exist.
- * `java.sql.SQLException` - The database is not reachable

• *RemoveProject*

```
public void RemoveProject( java.lang.String user, java.lang.String
hash, java.lang.String projectname )
```

– **Usage**

- * This method should be called to remove a project from the SPINGRID system. If there are job providers who were only allowed to provide jobs for the given project then they will loose the role of job provider. If the project admin was only project admin of the given project, he will loose the role of project admin.

– **Parameters**

- * **projectname** - The name of the project that will be removed

– **Exceptions**

- * **exceptions.ProjectException** - The specified project does not exist.
- * **exceptions.RoleException** - The specified user does not have the SystemAdmin role.
- * **java.sql.SQLException** - The database is not reachable
- * **exceptions.UserException** - The user does not exist

• *RemoveProjectAdmin*

```
public void RemoveProjectAdmin( java.lang.String user,  
java.lang.String hash, java.lang.String username, java.lang.String  
projectname )
```

– **Usage**

- * This method should be called to remove a project admin from the SPINGRID system.

– **Parameters**

- * **username** - The project admin that will be removed
- * **projectname** - The project where the project admin will be removed from

– **Exceptions**

- * **exceptions.ProjectException** - The specified project does not exist.
- * **exceptions.TrustException** - The SystemAdmin does not trust the target user.
- * **exceptions.RoleException** - The user does not have the role SystemAdmin or the target user does not have the role ProjectAdmin in the specified project.
- * **exceptions.UserException** - The user or target user does not exist.
- * **java.sql.SQLException** - The database is not reachable

• *RemoveUserForSystemAdmin*

```
public void RemoveUserForSystemAdmin( java.lang.String user,  
java.lang.String hash, java.lang.String username )
```

– **Parameters**

- * **user** - The user that wants to remove a user.
- * **hash** - The authentication hash.
- * **username** - The username of the user to be removed.

– **Exceptions**

- * **java.sql.SQLException** - Connection to the database failed.
- * **exceptions.UserException** - When either of the users does not exist.
- * **exceptions.AuthenticationException** - Hash failed.
- * **exceptions.RoleException** - When the user does not have the role systemadmin.

INTERFACE **IDUserCommands**

3795 The methods in this interface correspond to the commands a User can send (using the Client) to the dispatcher.

DECLARATION

`public interface IDUserCommands`

3800

METHODS

• *AddUser*

```
public void AddUser( java.lang.String user, java.lang.String hash,  
3805 java.lang.String username, java.lang.String password )
```

– Parameters

- * **user** - The user that wants to add a user.
- * **hash** - The authentication hash.
- * **username** - The username of the new user.
- * **password** - The password for the new user.

3810

– Exceptions

- * **java.sql.SQLException** - Connection to the database failed.
- * **exceptions.UserException** - User does not exist or target user already exists.
- * **exceptions.AuthenticationException** - Hash failed.
- * **exceptions.RoleException** - The user does not have the role ProjectAdmin or SystemAdmin

3815

• *infoApplication*

```
public String infoApplication( java.lang.String user, java.lang.String  
3820 hash, java.lang.String applicationname )
```

– Parameters

- * **user** - The user that sent the command
- * **hash** - The authentication hash
- * **applicationname** - The name of the application of which info is requested

3825

– Returns - Information about the application

– Exceptions

- * **exceptions.AuthenticationException** - Hash failed.
- * **exceptions.UserException** - User does not exist.
- * **java.sql.SQLException** - Connection to the database failed.
- * **exceptions.RoleException** - You do not have a sufficient role
- * **exceptions.ApplicationException** - The application does not exist.

3830

- *infoDataSet*

3835 public String infoDataSet(java.lang.String user, java.lang.String
 hash, java.lang.String datasetname)

- Parameters

- * user - The user that sent the command
 - * hash - The authentication hash
 - 3840 * datasetname - The name of the dataset of which info is requested

- Returns - Information about the dataset

- Exceptions

- * exceptions.AuthenticationException - Hash failed.
 - * exceptions.UserException - User does not exist.
 - 3845 * java.sql.SQLException - Connection to the database failed.
 - * exceptions.RoleException - You do not have a sufficient role
 - * exceptions.DataSetException - The dataset does not exist.
-

- *infoJob*

3850 public String infoJob(java.lang.String user, java.lang.String hash,
 java.lang.String jobnumber)

- Parameters

- * user - The user that sent the command
 - * hash - The authentication hash
 - 3855 * jobnumber - The identification of the job

- Returns - Information about the job

- Exceptions

- * exceptions.AuthenticationException - Hash failed.
 - * exceptions.UserException - User does not exist.
 - 3860 * java.sql.SQLException - Connection to the database failed.
 - * exceptions.RoleException - You do not have a sufficient role
 - * exceptions.JobException - The job does not exist.
-

- *infoProject*

3865 public String infoProject(java.lang.String user, java.lang.String
 hash, java.lang.String projectname)

- Parameters

- * user - The user that sent the command
 - * hash - The authentication hash
 - 3870 * projectname - The name of the project of which info is requested

- Returns - Information about the project

- Exceptions

- * exceptions.AuthenticationException - Hash failed.
 - * exceptions.UserException - User does not exist.
 - 3875 * java.sql.SQLException - Connection to the database failed.
 - * exceptions.RoleException - You do not have a sufficient role
 - * exceptions.ProjectException - The project does not exist.
-

- *infoUser*

```
3880 public String infoUser( java.lang.String user, java.lang.String hash,  
java.lang.String username )
```

- **Parameters**

- * **user** - The user that sent the command
 - * **hash** - The authentication hash
 - * **username** - The name of the user of which info is requested

- **Returns** - Information about the user

- **Exceptions**

- * **exceptions.AuthenticationException** - Hash failed.
 - * **exceptions.UserException** - User does not exist or target user does not exist.
 - * **java.sql.SQLException** - Connection to the database failed.
 - * **exceptions.RoleException** - You do not have a sufficient role

- *listUsers*

```
3895 public String listUsers( java.lang.String user, java.lang.String hash  
)
```

- **Parameters**

- * **user** - The user that wants to add a user.
 - * **hash** - The authentication hash.
 - * **password** - The password for the new user.

- **Exceptions**

- * **java.sql.SQLException** - Connection to the database failed.
 - * **exceptions.UserException** - User does not exist
 - * **exceptions.AuthenticationException** - Hash failed.
 - * **exceptions.RoleException** - The user does not have the role ProjectAdmin or SystemAdmin

- *SetPassword*

```
3910 public void SetPassword( java.lang.String username, java.lang.String  
hash, java.lang.String newpassword )
```

- **Parameters**

- * **username** - Name of the user that wants to set password
 - * **hash** - Authentication hash
 - * **newpassword** - The new desired password

- **Exceptions**

- * **exceptions.AuthenticationException** - When the hash is incorrect.
 - * **java.sql.SQLException** - When the database is unreachable.
 - * **exceptions.UserException** - When the user does not exist.

3.8 Package cTranslator

<i>Package Contents</i>	<i>Page</i>
-------------------------	-------------

3920

Classes	
CTranslator	120
<i>When a method in the Translator is called, it translates the parameters of the method into a GSP.</i>	

3.8.1 Classes

CLASS CTranslator

3925

When a method in the Translator is called, it translates the parameters of the method into a GSP. The GSPs are then send to the Communicator. GRPs from the Communicator are evaluated so that the appropriate message can be send to the user.

DECLARATION

3930

```
public class CTranslator

extends java.lang.Object
implements interfaces.ICUserCommands, interfaces.ICProjectAdminCommands,
interfaces.ICSystemAdminCommands, interfaces.ICJobProviderCommands,
interfaces.ICApplicationProviderCommands, interfaces.ICDataProviderCommands
```

CONSTRUCTORS

3935

- *CTranslator*

```
public CTranslator( )
```

- Usage

- * Creates a new instance of the CCommunicator object

METHODS

3940

- *begin*

```
public void begin( )
```

- Usage

- * Method that is called when an instance has been created and the interfaces have been set

3945

- *sendAddApplication*

```
public void sendAddApplication( java.lang.String application,
java.lang.String applicationname, java.lang.String description )
```

3950

- *sendAddApplicationProvider*

```
public void sendAddApplicationProvider( java.lang.String username )
```

- *sendAddData*

3955

```
public void sendAddData( java.net.URL [] data, java.lang.String
name, java.lang.String description )
```

- *sendAddDataProvider*
3960 public void sendAddDataProvider(java.lang.String username)
- *sendAddProject*
 public void sendAddProject(java.lang.String projectname,
 java.lang.String username)
- 3965 • *sendAddProjectAdmin*
 public void sendAddProjectAdmin(java.lang.String username,
 java.lang.String projectname)
- *sendAddUser*
3970 public void sendAddUser(java.lang.String username, java.lang.String
 password)
- *sendAllowOwnData*
 public void sendAllowOwnData(java.lang.String projectname,
3975 java.lang.String jobprovider)
- *sendApproveApplication*
 public void sendApproveApplication(java.lang.String projectname,
3980 java.lang.String application)
- *sendApproveApplicationProvider*
 public void sendApproveApplicationProvider(java.lang.String
 projectname, java.lang.String applicationprovider)
- 3985 • *sendApproveApplicationToProject*
 public void sendApproveApplicationToProject(java.lang.String
 application, java.lang.String projectname)
- *sendApproveDataProvider*
3990 public void sendApproveDataProvider(java.lang.String projectname,
 java.lang.String dataprovider)
- *sendApproveDataToProject*
 public void sendApproveDataToProject(java.lang.String data,
3995 java.lang.String projectname)
- *sendApproveJobProvider*
4000 public void sendApproveJobProvider(java.lang.String username,
 java.lang.String projectname)
- *sendDisallowOwnData*
 public void sendDisallowOwnData(java.lang.String projectname,
 java.lang.String jobprovider)
- 4005 • *sendDisapproveApplication*
 public void sendDisapproveApplication(java.lang.String projectname,
 java.lang.String application)

- *sendDisapproveApplicationProvider*
4010 public void sendDisapproveApplicationProvider(java.lang.String
 projectname, java.lang.String applicationprovider)
- *sendDisapproveApplicationToProject*
 public void sendDisapproveApplicationToProject(java.lang.String
4015 application, java.lang.String projectname)
- *sendDisapproveDataProvider*
 public void sendDisapproveDataProvider(java.lang.String
4020 projectname, java.lang.String dataprovider)
- *sendDisapproveDataToProject*
 public void sendDisapproveDataToProject(java.lang.String data,
 java.lang.String projectname)
- *sendDisapproveJobProvider*
4025 public void sendDisapproveJobProvider(java.lang.String username,
 java.lang.String projectname)
- *sendInfoApplication*
4030 public void sendInfoApplication(java.lang.String applicationname)
- *sendInfoDataSet*
 public void sendInfoDataSet(java.lang.String datasetname)
- *sendInfoJob*
4035 public void sendInfoJob(java.lang.String jobnumber)
- *sendInfoProject*
 public void sendInfoProject(java.lang.String projectname)
4040
- *sendInfoUser*
 public void sendInfoUser(java.lang.String username)
- *sendListApplicationProvidersForProjectAdmin*
4045 public void sendListApplicationProvidersForProjectAdmin()
- *sendListApplicationProvidersForSystemAdmin*
 public void sendListApplicationProvidersForSystemAdmin()
- *sendListApplicationsForApplicationProvider*
4050 public void sendListApplicationsForApplicationProvider()
- *sendListApplicationsForJobProvider*
 public void sendListApplicationsForJobProvider()
4055
- *sendListApplicationsForJobProvider*
 public void sendListApplicationsForJobProvider(java.lang.String
 project)
- *sendListApplicationsForProjectAdmin*
4060 public void sendListApplicationsForProjectAdmin()

- *sendListApplicationsForProjectAdmin*

- *sendListApplicationsForSystemAdmin*

- *sendListApplicationsForSystemAdmin*

- *sendListDataProvidersForProjectAdmin*

- *sendListDataProvidersForSystemAdmin*

- *sendListDataSetsForDataProvider*

- *sendListDataSetsForJobProvider*

- *sendListDataSetsForJobProvider*

- *sendListDataSetsForProjectAdmin*

- *sendListDataSetsForProjectAdmin*

- *sendListDataSetsForSystemAdmin*

- *sendListDataSetsForSystemAdmin*

- *sendListJobProvidersForProjectAdmin*

- *sendListJobProvidersForProjectAdmin*

- *sendListJobProvidersForSystemAdmin*

- *sendListJobProvidersForSystemAdmin*

```

4120 • sendListJobsForApplicationProvider
      public void sendListJobsForApplicationProvider( )

      • sendListJobsForApplicationProvider
        public void sendListJobsForApplicationProvider( java.lang.String
          project )

4125 • sendListJobsForJobProvider
      public void sendListJobsForJobProvider( )

      • sendListJobsForJobProvider
        public void sendListJobsForJobProvider( java.lang.String project )

4130 • sendListJobsForProjectAdmin
      public void sendListJobsForProjectAdmin( )

      • sendListJobsForProjectAdmin
        public void sendListJobsForProjectAdmin( java.lang.String project )

4135 • sendListJobsForSystemAdmin
      public void sendListJobsForSystemAdmin( )

      • sendListJobsForSystemAdmin
        public void sendListJobsForSystemAdmin( java.lang.String project )

4140 • sendListProjectAdminsForSystemAdmin
      public void sendListProjectAdminsForSystemAdmin( )

4145 • sendListProjectsForApplicationProvider
      public void sendListProjectsForApplicationProvider( )

      • sendListProjectsForDataProvider
        public void sendListProjectsForDataProvider( )

4150 • sendListProjectsForJobProvider
      public void sendListProjectsForJobProvider( )

      • sendListProjectsForProjectAdmin
        public void sendListProjectsForProjectAdmin( )

4155 • sendListProjectsForSystemAdmin
      public void sendListProjectsForSystemAdmin( )

4160 • sendListUsers
      public void sendListUsers( )

      • sendOfferJob
        public void sendOfferJob( java.lang.String projectname,
          java.lang.String job )

4165 • sendRemoveApplication
      public void sendRemoveApplication( java.lang.String application )

4170

```

- *sendRemoveApplicationProvider*
`public void sendRemoveApplicationProvider(java.lang.String
username)`

- 4175 • *sendRemoveData*
`public void sendRemoveData(java.lang.String data)`

- *sendRemoveDataProvider*
4180 `public void sendRemoveDataProvider(java.lang.String username)`

- *sendRemoveJobForJobProvider*
`public void sendRemoveJobForJobProvider(java.lang.String job)`

- *sendRemoveJobForProjectAdmin*
4185 `public void sendRemoveJobForProjectAdmin(java.lang.String job)`

- *sendRemoveProject*
`public void sendRemoveProject(java.lang.String projectname)`

- 4190 • *sendRemoveProjectAdmin*
`public void sendRemoveProjectAdmin(java.lang.String username,
java.lang.String projectname)`

- *sendRemoveUserForSystemAdmin*
4195 `public void sendRemoveUserForSystemAdmin(java.lang.String
username)`

- *sendSetPassword*
4200 `public void sendSetPassword(java.lang.String password)`

- *setInterfaces*
`public void setInterfaces(interfaces.ICommunicator icCommunicator,
interfaces.ICOutputProcessor icOutputProcessor)`
 - Usage
4205 * Sets references to all interfaces this component might need
 - Parameters
 - * icCommunicator -
 - * icOutputProcessor -

3.9 Package aJobExecutor

Package Contents *Page*

4210

Classes

AJobExecutor 127

This component executes jobs.

3.9.1 Classes

CLASS **AJobExecutor**

4215

This component executes jobs. A signal will be send to the JobScheduler when a job is done executing.

DECLARATION

4220

```
public class AJobExecutor
extends java.lang.Object
implements interfaces.IAJobExecutor
```

FIELDS

4225

- public static final int STATUS_SUCCESS
—
- public static final int STATUS_FAILED
—

CONSTRUCTORS

4230

- *AJobExecutor*
public **AJobExecutor**()
— **Usage**
* Creates a new instance of the CCommunicator object

METHODS

4235

- *begin*
public void **begin**()
— **Usage**
* Method that is called when an instance has been created and the interfaces have been set

4240

-
- *setInterfaces*
public void **setInterfaces**(interfaces.IAJobSchedulerForJobExecutor
iaJobScheduler)

- 4245 – **Usage**
 * Sets references to all interfaces this component might need
- **Parameters**
 * `iaJobScheduler` -
-
- 4250 • *startJob*
 public int **startJob**(java.lang.String **path**, java.lang.String []
 myproglis)
-
- 4255 • *stopJob*
 public void **stopJob**(int **jobid**)

3.10 Package dDatabaseManager

Package Contents *Page*

Classes

dDatabaseManager 130

The Database Manager has several functions: - Responsible of all the communication with the SQL-database.

3.10.1 Classes

CLASS DDatabaseManager

The Database Manager has several functions: - Responsible of all the communication with the SQL-database. - Perform update/read requests of the Client Manager and the Agent Manager.

DECLARATION

```
public class DDatabaseManager
extends java.lang.Object
implements interfaces.IDQueryHandler
```

METHODS

- *begin*
public void **begin**()
 - **Usage**
 - * Begin. Not used.
- *clone*
public Object **clone**()
 - **Usage**
 - * Since this is a singleton object, it may not be cloned.
 - **Returns** - This method never returns.
 - **Exceptions**
 - * `java.lang.CloneNotSupportedException` - Always throws `CloneNotSupportedException`.
- *connectToDatabase*
public Connection **connectToDatabase**()
 - **Usage**
 - * Create a new connection to the database.
 - **Returns** - The newly created connection.
 - **Exceptions**
 - * `java.sql.SQLException` - Thrown if connecting fails.
- *getInstance*
public static DDatabaseManager **getInstance**()

- 4295 – **Usage**
 * The public method to get a reference to the DDatabaseManager
 instantiation (or create one, if it doesn't exist yet).
- **Returns** - A reference to the DDatabaseManager
-
- 4300 • *PerformQuery*
 public ResultSet **PerformQuery**(java.lang.String query)
- **Usage**
 * Perform a query on the database, and return the result in a ResultSet.
- **Parameters**
4305 * query - A string containing the SQL query.
- **Returns** - A Resultset containing the query results.
- **Exceptions**
 * java.sql.SQLException - Thrown if the query or connecting to the
 database fails.
-
- 4310 • *PerformUpdate*
 public int **PerformUpdate**(java.lang.String query)
- **Usage**
 * Perform an update on the database, and return the number of rows affected.
- 4315 – **Parameters**
 * query - A string containing the SQL query.
- **Returns** - An integer representing the number of rows accepted.
- **Exceptions**
 * java.sql.SQLException - Thrown if the query or connecting to the
4320 database fails.
-
- *setConnectInfo*
 public void **setConnectInfo**(java.lang.String dbUrl, java.lang.String
 dbUser, java.lang.String dbPassword)
- 4325 – **Usage**
 * Set the database location and login credentials.
- **Parameters**
 * dbUrl - URL to the database.
 * dbUser - Username to provide to the database.
4330 * dbPassword - Password to provide to the database.
-
- *setInterfaces*
 public void **setInterfaces**()
- **Usage**
4335 * Set the interfaces that are used by this class. Not used.

3.11 Package JSDL

Package Contents

Page

Classes

	JSDL	133
	<i>This class can parse a JSDL document, making its contents accessible for easy inspection and modification.</i>	
	JSDLApplication	135
4340	<i>This class represents the JSDLApplication.</i>	
	JSDLComplexRange	137
	<i>This class represents the JSDLComplexRange.</i>	
	JSDLDataStaging	139
	<i>This class represents the JSDLDataStaging.</i>	
4345	JSDLDataStagingList	142
	<i>This class represents the JSDLDataStagingList (a list of JSDLDataStagings).</i>	
	JSDLFileSystem	143
	<i>This class represents the JSDLFileSystem.</i>	
	JSDLJobIdentification	144
4350	<i>This class represents the JSDLJobIdentification.</i>	
	JSDLLowerBoundedRange	146
	<i>This class represents the JSDLLowerBoundedRange.</i>	
	JSDLOperatingSystem	147
	<i>This class represents the JSDLOperatingSystem.</i>	
4355	JSDLResources	149
	<i>This class represents the JSDLResources.</i>	
	JSDLSimpleRange	154
	<i>This class represents the JSDLSimpleRange.</i>	
	JSDLSpingridApplication	156
4360	<i>This class represents the JSDLSpingridApplication.</i>	
	JSDLSpingridDatasets	158
	<i>This class represents the JSDLSpingridDatasets.</i>	
	JSDLSpingridExecutable	159
	<i>This class represents the JSDLSpingridExecutable.</i>	
4365	JSDLSpingridOperatingSystem	161
	<i>This class represents the JSDLSpingridOperatingSystem.</i>	

3.11.1 Classes

CLASS **JSDL**

4370

This class can parse a JSDL document, making its contents accessible for easy inspection and modification. This class can also be re-exported to a JSDL document.

DECLARATION

4375

```
public class JSDL
extends java.lang.Object
```

FIELDS

4380

- public static final int OS_WINXP
 - Constant representing support for the Windows XP operating system.
- public static final int OS_LINUX
 - Constant representing support for the GNU/Linux operating system.
- public static final int OS_MACOS
 - Constant representing support for the Mac OS X operating system.
- 4385 • public static final String OS_NAMES
 - Constants for the operating system names.
- public static final String INDENT_STR
 - The tree and XML representation are indented using this string. The recommended value for this constant is " ".
- 4390 • public static String NEWLINE
 - Newline for handy access.

CONSTRUCTORS

4395

- *JSDL*
public **JSDL**(java.lang.String xmldoc)
 - **Usage**
 - * Take a JSDL document (in the form of a string), and parse it.
 - **Parameters**
 - * **xmldoc** - The JSDL document to parse.

4400 METHODS

• *getJSDLApplication*

```
public JSDLApplication getJSDLApplication( )
```

– **Usage**

* Get access to the JSDLApplication subpart.

– **Returns** - The JSDLApplication part of this JSDL document.

• *getJSDLDataStagingList*

```
public JSDLDataStagingList getJSDLDataStagingList( )
```

– **Usage**

* Get access to the JSDLDataStagingList subpart.

– **Returns** - The JSDLDataStagingList part of this JSDL document.

• *getJSDLJobIdentification*

```
public JSDLJobIdentification getJSDLJobIdentification( )
```

– **Usage**

* Get access to the JSDLJobIdentification subpart.

– **Returns** - The JSDLJobIdentification part of this JSDL document.

• *getJSDLResources*

```
public JSDLResources getJSDLResources( )
```

– **Usage**

* Get access to the JSDLResources subpart.

– **Returns** - The JSDLResources part of this JSDL document.

• *indentString*

```
public static String indentString( int indent )
```

– **Usage**

* Create a string to be used as indentation. Returns a concatenation of indent times the indentation string.

– **Parameters**

* **indent** - The amount of indentation desired.

– **Returns** - The complete indentation string.

• *treeRepresentation*

```
public String treeRepresentation( )
```

– **Usage**

* Return a multi-line string containing a tree representation of this JSDL document.

– **Returns** - The tree representation.

- *treeRepresentation*

public String treeRepresentation(int in)

- **Usage**

4445 * Return a multi-line string containing a tree representation of this JSDL document.

- **Parameters**

 * in - The amount of indenting to apply.

- **Returns** - The tree representation.

4450

- *xmlRepresentation*

public String xmlRepresentation()

- **Usage**

4455 * Return a multi-line string containing an XML representation of this JSDL document.

- **Returns** - The XML representation.

- *xmlRepresentation*

public String xmlRepresentation(int in)

- **Usage**

4460 * Return a multi-line string containing an XML representation of this JSDL document.

- **Parameters**

 * in - The amount of indenting to apply.

4465 * **Returns** - The XML representation.

CLASS JSDLApplication

This class represents the JSDLApplication.

DECLARATION

4470

```
public class JSDLApplication
    extends java.lang.Object
```

CONSTRUCTORS

- *JSDLApplication*

public JSDLApplication(org.w3c.dom.Node jobapplication)

- **Usage**
 - * Create the JSDLApplication, parsing a w3c dom Node.
- **Parameters**
 - * `jobapplication` - The w3c dom Node to parse.

4480

METHODS

- *getApplicationName*

public String **getApplicationName**()

4485

- **Usage**
 - * Get access to the application name.
- **Returns** - The application name of this JSDLApplication.

- *getApplicationVersion*

public String **getApplicationVersion**()

4490

- **Usage**
 - * Get access to the application version.
- **Returns** - The application version of this JSDLApplication.

- *getDescription*

public String **getDescription**()

4495

- **Usage**
 - * Get access to the application description.
- **Returns** - The application description of this JSDLApplication.

4500

- *getJSDLSpinggridApplication*

public JSDLSpinggridApplication **getJSDLSpinggridApplication**()

4505

- **Usage**
 - * Get access to the JSDLSpinggridApplication subpart.
- **Returns** - The JSDLSpinggridApplication part of this JSDLApplication.

- *getJSDLSpinggridDatasets*

public JSDLSpinggridDatasets **getJSDLSpinggridDatasets**()

4510

- **Usage**
 - * Get access to the JSDLSpinggridDatasets subpart.
- **Returns** - The JSDLSpinggridDatasets part of this JSDLApplication.

- *getMyProgList*

public String **getMyProgList**(java.lang.String aOS)

4515

- **Usage**

- * Returns a list of commands (executable + paramaters) for a given operating system, that can be executed directly by the jobExecutor

– **Parameters**

- * **aOS** - a name of an operating system

– **Returns** - String[] list of commands

• *treeRepresentation*

public String treeRepresentation(int in)

– **Usage**

- * Return a multi-line string containing a tree representation of this JSDLApplication.

– **Parameters**

- * **in** - The amount of indenting to apply.

– **Returns** - The tree representation.

• *xmlRepresentation*

public String xmlRepresentation(int in)

– **Usage**

- * Return a multi-line string containing an XML representation of this JSDLApplication.

– **Parameters**

- * **in** - The amount of indenting to apply.

– **Returns** - The XML representation.

CLASS JSDLComplexRange

This class represents the JSDLComplexRange.

DECLARATION

```
public class JSDLComplexRange
extends java.lang.Object
```

CONSTRUCTORS

• *JSDLComplexRange*

public JSDLComplexRange(org.w3c.dom.Node range)

– **Usage**

- * Create the JSDLComplexRange, parsing a w3c dom Node.
- **Parameters**
 - * **range** - The w3c dom Node to parse.

METHODS

4555

- *getRange*
`public JSDLSimpleRange getRange(int index)`

- **Usage**
 - * Get access to the simple range specified by index.
 - **Returns** - The requested simple range.
-

4560

- *getRangeCount*
`public int getRangeCount()`

- **Usage**
 - * Get the number of simple ranges in this complex range.
 - **Returns** - The number of ranges.
-

4565

- *isInRange*
`public boolean isInRange(double v)`

- **Usage**
 - * Check if a given value is within any of the simple ranges making up this complex range (the value is accepted), or not (the value is unacceptable).
 - **Parameters**
 - * **v** - The value to test.
 - **Returns** - True if v is in any of the ranges, false otherwise.
-

4570

- *toString*
`public String toString()`

- **Usage**
 - * Return a human-readable description of this complex range.
 - **Returns** - The description.
-

4580

- *xmlRepresentation*
`public String xmlRepresentation(int in)`

- **Usage**
 - * Return a multi-line string containing an XML representation of this JSDLComplexRange.
- **Parameters**
 - * **in** - The amount of indenting to apply.
- **Returns** - The XML representation.

4590

CLASS JSDLDataStaging

This class represents the JSDLDataStaging.

DECLARATION

4595

```
public class JSDLDataStaging
extends java.lang.Object
```

FIELDS

4600

- public static final String SOURCE
 - Constant representing the source keyword.
- public static final String TARGET
 - Constant representing the target keyword.
- public static final String FS_HOME
 - Constant representing the home filesystem name.
- public static final String FS_TMP
 - Constant representing the temporary filesystem name.

4605

CONSTRUCTORS

4610

- *JSDLDataStaging*
public **JSDLDataStaging**()
 - **Usage**
 - * Empty constructor. All fields are initialized to empty, they need to be explicitly set.

4615

-
- *JSDLDataStaging*
public **JSDLDataStaging**(org.w3c.dom.Node **datastage**)
 - **Usage**
 - * Create the JSDLDataStaging, parsing a w3c dom Node.
 - **Parameters**
 - * **datastage** - the w3c dom Node to parse.

4620

METHODS

• *getCreationFlag*

4625 public String getCreationFlag()

– Usage

* Get access to the creation flag.

– Returns - The creation flag of this JSDLDDataStaging.

• *getFileName*

4630 public String getFileName()

– Usage

* Get access to the filename.

– Returns - The filename of this JSDLDDataStaging.

• *getFileSystemName*

4635 public String getFileSystemName()

– Usage

* Get access to the filesystem name.

4640 – Returns - The filesystem name of this JSDLDDataStaging.

• *getType*

public String getType()

– Usage

4645 * Get access to the datastaging type.

– Returns - The type of this JSDLDDataStaging.

• *getUri*

public String getUri()

– Usage

* Get access to the URI.

4650 – Returns - The URI of this JSDLDDataStaging.

• *setCreationFlag*

4655 public void setCreationFlag(java.lang.String creationflag)

– Usage

* Set the creation flag of this JSDLDDataStaging.

– Parameters

* creationflag - The new creation flag.

• *setFileName*

4660 public void setFileName(java.lang.String filename)

– **Usage**
 * Set the filename of this JSDLDDataStaging.

4665 – **Parameters**
 * `filename` - The new filename.

• *setFileSystemName*
 public void **setFileSystemName**(java.lang.String `filesystemname`)

4670 – **Usage**
 * Set the filesystem name of this JSDLDDataStaging.

– **Parameters**
 * `filesystemname` - The new filesystem name.

4675 • *setType*
 public void **setType**(java.lang.String `type`)

– **Usage**
 * Set the type of this JSDLDDataStaging.

– **Parameters**
 4680 * `type` - The new type.

• *setUri*
 public void **setUri**(java.lang.String `uri`)

– **Usage**
 4685 * Set the URI of this JSDLDDataStaging.

– **Parameters**
 * `uri` - The new URI.

• *treeRepresentation*
 4690 public String **treeRepresentation**(int `in`)

– **Usage**
 * Return a multi-line string containing a tree representation of this JSDLDDataStaging.

– **Parameters**
 4695 * `in` - The amount of indenting to apply.

– **Returns** - The tree representation.

• *xmlRepresentation*
 public String **xmlRepresentation**(int `in`)

4700 – **Usage**
 * Return a multi-line string containing an XML representation of this JSDLDDataStaging.

– **Parameters**
 * `in` - The amount of indenting to apply.

4705 – **Returns** - The XML representation.

CLASS JSDLDataStagingList

This class represents the JSDLDataStagingList (a list of JSDLDataStagings).

DECLARATION

```
public class JSDLDataStagingList
extends java.lang.Object
```

CONSTRUCTORS

• *JSDLDataStagingList*

```
public JSDLDataStagingList( org.w3c.dom.NodeList  datastages )
```

– **Usage**

* Create the JSDLDataStagingList, parsing a w3c dom NodeList.

– **Parameters**

* **datastages** - the w3c dom NodeList to parse.

METHODS

• *getCount*

```
public int getCount( )
```

– **Usage**

* Get the number of datastagings in this datastaging list.

– **Returns** - The number of datastagings.• *getJSDLDataStagings*

```
public LinkedList getJSDLDataStagings( )
```

– **Usage**

* Get access to the actual list of datastagings of this JSDLDataStagingList.

– **Returns** - The list of JSDLDataStagings.• *treeRepresentation*

```
public String treeRepresentation( int  in )
```

– **Usage**

* Return a multi-line string containing a tree representation of this JSDLDataStagingList.

– **Parameters**

- * **in** - The amount of indenting to apply.
- **Returns** - The tree representation.

- *xmlRepresentation*

4745 public String xmlRepresentation(int in)

- **Usage**
 - * Return a multi-line string containing an XML representation of this JSDLDataStagingList.
- **Parameters**
 - 4750 * **in** - The amount of indenting to apply.
- **Returns** - The XML representation.

CLASS JSDLFileSystem

This class represents the JSDLFileSystem.

DECLARATION

public class JSDLFileSystem
extends java.lang.Object

CONSTRUCTORS

- *JSDLFileSystem*

4760 public **JSDLFileSystem**(org.w3c.dom.Node filesystem)

 - **Usage**
 - * Create the JSDLFileSystem, parsing a w3c dom Node.
 - 4765 – **Parameters**
 - * **filesystem** - The w3c dom Node to parse.

METHODS

- *getDescription*

4770 public String **getDescription**()

 - **Usage**
 - * Get access to the filesystem description.
 - **Returns** - The description of this JSDLFileSystem.

- 4775 • *getDiskSpace*
public JSDLLowerBoundedRange **getDiskSpace**()
 - **Usage**
 - * Get access to the JSDLLowerBoundedRange subpart.
 - **Returns** - The JSDLLowerBoundedRange part of this JSDLFileSystem.
- 4780 • *getFileSystemType*
public String **getFileSystemType**()
 - **Usage**
 - * Get access to the filesystem type.
 - 4785 – **Returns** - The filesystem type of this JSDLFileSystem.
- *getName*
public String **getName**()
 - **Usage**
 - 4790 * Get access to the filesystem name.
 - **Returns** - The filesystem name of this JSDLFileSystem.
- *treeRepresentation*
public String **treeRepresentation**(int in)
 - 4795 – **Usage**
 - * Return a multi-line string containing a tree representation of this JSDLFileSystem.
 - **Parameters**
 - * in - The amount of indenting to apply.
 - 4800 – **Returns** - The tree representation.
- *xmlRepresentation*
public String **xmlRepresentation**(int in)
 - **Usage**
 - 4805 * Return a multi-line string containing an XML representation of this JSDLFileSystem.
 - **Parameters**
 - * in - The amount of indenting to apply.
 - **Returns** - The XML representation.

4810 CLASS JSDLJobIdentification

This class represents the JSDLJobIdentification.

DECLARATION

```
public class JSDLJobIdentification
4815 extends java.lang.Object
```

CONSTRUCTORS

• *JSDLJobIdentification*

```
4820 public JSDLJobIdentification( org.w3c.dom.Node jobidentification )
```

– **Usage**

* Create the JSDLJobIdentification, parsing a w3c dom Node.

– **Parameters**

* jobidentification - the w3c dom Node to parse.

4825 **METHODS**

• *getDescription*

```
public String getDescription( )
```

– **Usage**

* Get access to the job description.

– **Returns** - The job description of this JSDLJobIdentification.

• *getJobAnnotation*

```
public String getJobAnnotation( )
```

– **Usage**

* Get access to the job annotation.

– **Returns** - The job annotation of this JSDLJobIdentification.

• *getJobName*

```
4840 public String getJobName( )
```

– **Usage**

* Get access to the job name.

– **Returns** - The job name of this JSDLJobIdentification.

• *getJobProject*

```
4845 public String getJobProject( )
```

– **Usage**

* Get access to the job's project.

– **Returns** - The job's project of this JSDLJobIdentification.

4850

- *treeRepresentation*

public String treeRepresentation(int in)

- **Usage**

- * Return a multi-line string containing a tree representation of this JSDLJobIdentification.

4855

- **Parameters**

- * **in** - The amount of indenting to apply.

- **Returns** - The tree representation.

4860

- *xmlRepresentation*

public String xmlRepresentation(int in)

- **Usage**

- * Return a multi-line string containing an XML representation of this JSDLJobIdentification.

4865

- **Parameters**

- * **in** - The amount of indenting to apply.

- **Returns** - The XML representation.

CLASS JSDLLowerBoundedRange

4870 This class represents the JSDLLowerBoundedRange.

DECLARATION

<pre>public class JSDLLowerBoundedRange extends java.lang.Object</pre>

4875

CONSTRUCTORS

- *JSDLLowerBoundedRange*

public JSDLLowerBoundedRange(org.w3c.dom.Node range)

- **Usage**

- * Create the JSDLLowerBoundedRange, parsing a w3c dom Node.

4880

- **Parameters**

- * **range** - the w3c dom Node to parse.

METHODS

- 4885 • *getLowerBound*
 public double **getLowerBound**()
 – **Usage**
 * Get the actual lowerbound.
 – **Returns** - The lowerbound.
-
- 4890 • *isInRange*
 public boolean **isInRange**(double v)
 – **Usage**
 * Test whether or not a given value qualifies for this range.
 – **Parameters**
 * v - The value to test.
 – **Returns** - True if v is at least the lower bound (or there's no lowerbound set).
-
- 4900 • *toString*
 public String **toString**()
 – **Usage**
 * Get a human-readable description of this lower bound.
 – **Returns** - The description.
-
- 4905 • *xmlRepresentation*
 public String **xmlRepresentation**(int in)
 – **Usage**
 * Return a multi-line string containing an XML representation of this
 JSDLLowerBoundedRange.
 – **Parameters**
 * in - The amount of indenting to apply.
 – **Returns** - The XML representation.

CLASS **JSDLOperatingSystem**

4915 This class represents the JSDLOperatingSystem.

DECLARATION

```
public class JSDLOperatingSystem
extends java.lang.Object
```

4920 **CONSTRUCTORS**

• *JSDLOperatingSystem*

```
public JSDLOperatingSystem( org.w3c.dom.Node node )
```

– **Usage**

4925 * Create the JSDLOperatingSystem, parsing a w3c dom Node.

– **Parameters** * **node** - the w3c dom Node to parse.

METHODS

4930 • *getOSDescription*

```
public String getOSDescription( )
```

– **Usage**

* Get access to the operating system description.

– **Returns** - The operating system description of this JSDLOperatingSystem.

4935

• *getOSName*

```
public String getOSName( )
```

– **Usage**

* Get access to the operating system name.

– **Returns** - The operating system name of this JSDLOperatingSystem.

4940

• *getOSVersion*

```
public String getOSVersion( )
```

– **Usage**

* Get access to the operating system version.

– **Returns** - The operating system version of this JSDLOperatingSystem.

4945

• *setOSDescription*

```
public void setOSDescription( java.lang.String aOSDescription )
```

– **Usage**

* Set the operating system description of this JSDLOperatingSystem.

– **Parameters** * **aOSDescription** - The new operating system description.

4950

4955 • *setOSName*

```
public void setOSName( java.lang.String aOSName )
```

– **Usage**

* Set the operating system name of this JSDLOperatingSystem.

– **Parameters**

4960 * **aOSName** - The new operating system name.

• *setOSVersion*

public void setOSVersion(java.lang.String **aOSVersion**)

– **Usage**

4965 * Set operating system version the of this JSDLOperatingSystem.

– **Parameters**

 * **aOSVersion** - The new operating system version.

• *treeRepresentation*

4970 public String treeRepresentation(int **in**)

– **Usage**

 * Return a multi-line string containing a tree representation of this JSDLOperatingSystem.

– **Parameters**

4975 * **in** - The amount of indenting to apply.

– **Returns** - The tree representation.

• *xmlRepresentation*

public String xmlRepresentation(int **in**)

– **Usage**

4980 * Return a multi-line string containing an XML representation of this JSDLOperatingSystem.

– **Parameters**

 * **in** - The amount of indenting to apply.

4985 – **Returns** - The XML representation.

CLASS JSDLResources

This class represents the JSDLResources.

DECLARATION

4990

```
public class JSDLResources
extends java.lang.Object
```

CONSTRUCTORS

- 4995 • *JSDLResources*
 public **JSDLResources**(org.w3c.dom.Node resources)
- **Usage**
 * Create the JSDLResources, parsing a w3c dom Node.
- **Parameters**
5000 * **resources** - The w3c dom Node to parse.

METHODS

- *getCPUAarchitectureName*
 public String **getCPUAarchitectureName**()
- 5005 – **Usage**
 * Get access to the cpu architecture name.
- **Returns** - The cpu architecture name of this JSDLResources.
-
- *getExclusiveExecution*
5010 public boolean **getExclusiveExecution**()
- **Usage**
 * Get the exclusive execution flag.
- **Returns** - The exclusive execution flag of this JSDLResources.
-
- 5015 • *getFileSystems*
 public LinkedList **getFileSystems**()
- **Usage**
 * Get access to the list of JSDLFileSystem subparts.
- **Returns** - The list of JSDLFileSystem parts of this JSDLResources.
-
- 5020 • *getIndividualCPUCount*
 public JSDLLowerBoundedRange **getIndividualCPUCount**()
- **Usage**
 * Get access to the minimum CPU count subpart.
- 5025 – **Returns** - The minimum CPU count part of this JSDLResources.
-
- *getIndividualCPUSpeed*
 public JSDLLowerBoundedRange **getIndividualCPUSpeed**()
- **Usage**
5030 * Get access to the minimum CPU speed subpart.
- **Returns** - The minimum CPU speed part of this JSDLResources.

- *getIndividualDiskSpace*

public JSDDLLowerBoundedRange **getIndividualDiskSpace**()

– **Usage**

* Get access to the minimum disk space subpart.

– **Returns** - The minimum disk space part of this JSDDLResources.

- *getIndividualNetworkBandwidth*

public JSDDLLowerBoundedRange **getIndividualNetworkBandwidth**()

– **Usage**

* Get access to the minimum network bandwidth subpart.

– **Returns** - The minimum network bandwidth part of this JSDDLResources.

- *getIndividualPhysicalMemory*

public JSDDLLowerBoundedRange **getIndividualPhysicalMemory**()

– **Usage**

* Get access to the minimum physical memory subpart.

– **Returns** - The minimum physical memory part of this JSDDLResources.

- *getIndividualVirtualMemory*

public JSDDLLowerBoundedRange **getIndividualVirtualMemory**()

– **Usage**

* Get access to the minimum virtual memory subpart.

– **Returns** - The minimum virtual memory part of this JSDDLResources.

- *getOS*

public JSDDLOperatingSystem **getOS**()

– **Usage**

* Get access to the JSDDLOperatingSystem subpart.

– **Returns** - The JSDDLOperatingSystem part of this JSDDLResources.

- *setCPUAarchitectureName*

public void **setCPUAarchitectureName**(java.lang.String aName)

– **Usage**

* Set the CPU architecture of this JSDDLResources.

– **Parameters**

* aName - The new CPU architecture.

- *setExclusiveExecution*

public void **setExclusiveExecution**(boolean ee)

– **Usage**

* Set the exclusive execution flag of this JSDLResources.

– **Parameters**

* **ee** - The new exclusive execution flag.

• *setFileSystems*

public void **setFileSystems**(JSDL.JSDLFileSystem **afilesystem**)

– **Usage**

* Set the list of JSDLFileSystem subparts of this JSDLResources.

– **Parameters**

* **afilesystem** - The new list of JSDLFileSystem subparts.

• *setIndividualCPUCount*

public void **setIndividualCPUCount**(JSDL.JSDLLowerBoundedRange **count**)

– **Usage**

* Set the minimum CPU count of this JSDLResources.

– **Parameters**

* **count** - The new minimum CPU count.

• *setIndividualCPUSpeed*

public void **setIndividualCPUSpeed**(JSDL.JSDLLowerBoundedRange **range**)

– **Usage**

* Set the minimum CPU speed of this JSDLResources.

– **Parameters**

* **range** - The new minimum CPU speed.

• *setIndividualDiskSpace*

public void **setIndividualDiskSpace**(JSDL.JSDLLowerBoundedRange **diskspace**)

– **Usage**

* Set the minimum diskspace of this JSDLResources.

– **Parameters**

* **diskspace** - The new minimum diskspace.

• *setIndividualNetworkBandwidth*

public void **setIndividualNetworkBandwidth**(JSDL.JSDLLowerBoundedRange **bandwidth**)

– **Usage**

* Set the minimum network bandwidth of this JSDLResources.

– **Parameters**

* **bandwidth** - The new minimum network bandwidth.

5115

- *setIndividualPhysicalMemory*
 public void **setIndividualPhysicalMemory**(JSDL.JSDDLLowerBoundedRange
 memory)
 – **Usage**
 * Set the minimum physical memory of this JSDLResources.
 5120
 – **Parameters**
 * **memory** - The new minimum physical memory.

- *setIndividualVirtualMemory*
 5125 public void **setIndividualVirtualMemory**(JSDL.JSDDLLowerBoundedRange
 memory)
 – **Usage**
 * Set the minimum virtual memory of this JSDLResources.
 – **Parameters**
 5130 * **memory** - The new minimum virtual memory.

- *setOS*
 public void **setOS**(JSDL.JSDDLOperatingSystem aOS)
 – **Usage**
 5135 * Set the operating system of this JSDLResources.
 – **Parameters**
 * **aOS** - The new operating system.

- *treeRepresentation*
 5140 public String **treeRepresentation**(int in)
 – **Usage**
 * Return a multi-line string containing a tree representation of this
 JSDLResources.
 – **Parameters**
 5145 * **in** - The amount of indenting to apply.
 – **Returns** - The tree representation.

- *xmlRepresentation*
 public String **xmlRepresentation**(int in)
 5150
 – **Usage**
 * Return a multi-line string containing an XML representation of this
 JSDLResources.
 – **Parameters**
 * **in** - The amount of indenting to apply.
 5155
 – **Returns** - The XML representation.

CLASS JSDLSimpleRange

This class represents the JSDLSimpleRange.

DECLARATION

5160

```
public class JSDLSimpleRange
extends java.lang.Object
```

FIELDS

5165

- **public static final int TYPE_INVALID**
 - Constant representing that this range is not valid.
- **public static final int TYPE_LOWERBOUND**
 - Constant representing that this range is a lowerbound.
- **public static final int TYPE_UPPERBOUND**
 - Constant representing that this range is an upperbound.
- **public static final int TYPE_EXACT**
 - Constant representing that this range is an exact value.
- **public static final int TYPE_INTERVAL**
 - Constant representing that this range is an interval.

5170

5175

CONSTRUCTORS

- *JSDLSimpleRange*
public JSDLSimpleRange(org.w3c.dom.Node range)
 - **Usage**
 - * Create the JSDLSimpleRange, parsing a w3c dom Node.
 - **Parameters**
 - * **range** - The w3c dom Node to parse.

5180

METHODS

-
- 5185 • *getEpsilon*
 public double **getEpsilon**()
- **Usage**
 * Get the allowed deviation.
- **Returns** - The allowed deviation of this JSDLSimpleRange.
-
- 5190 • *getExact*
 public double **getExact**()
- **Usage**
 * Get the exact value.
- 5195 – **Returns** - The exact value of this JSDLSimpleRange.
-
- *getLowerBound*
 public double **getLowerBound**()
- **Usage**
 * Get the lower bound.
- 5200 – **Returns** - The lower bound of this JSDLSimpleRange.
-
- *getLowerExclusive*
 public boolean **getLowerExclusive**()
- 5205 – **Usage**
 * Get the lower bound exclusive flag.
- **Returns** - The lower bound exclusive flag of this JSDLSimpleRange.
-
- *getType*
 public int **getType**()
- 5210 – **Usage**
 * Get the type.
- **Returns** - The type of this JSDLSimpleRange.
-
- 5215 • *getUpperBound*
 public double **getUpperBound**()
- **Usage**
 * Get the upper bound.
- **Returns** - The upper bound of this JSDLSimpleRange.
-
- 5220 • *getUpperExclusive*
 public boolean **getUpperExclusive**()
- **Usage**

5225 * Get the upper bound exclusive flag.
 – **Returns** - The upper bound exclusive flag of this JSDDLSimpleRange.

• *isInRange*
 public boolean **isInRange**(double v)

 – **Usage**
 * Check if a given value is within this range (the value is accepted), or not (the
 value is unacceptable).

5230 – **Parameters**
 * v - The value to test.

 – **Returns** - True if v in this range, false otherwise.

5235 • *toString*
 public String **toString**()

 – **Usage**
 * Return a human-readable description of this complex range.

5240 – **Returns** - The description.

• *xmlRepresentation*
 public String **xmlRepresentation**(int in)

 – **Usage**
 * Return a multi-line string containing an XML representation of this
 JSDDLSimpleRange.

5245 – **Parameters**
 * in - The amount of indenting to apply.

 – **Returns** - The XML representation.

5250 **CLASS JSDDLSpingridApplication**

This class represents the JSDDLSpingridApplication.

DECLARATION

5255 public class JSDDLSpingridApplication
 extends java.lang.Object

CONSTRUCTORS

- *JSDLSpingridApplication*

5260 public **JSDLSpingridApplication**(org.w3c.dom.Node application)

- **Usage**

- * Create the JSDLSpingridApplication, parsing a w3c dom Node.

- **Parameters**

- * application - The w3c dom Node to parse.

5265

- *JSDLSpingridApplication*

public **JSDLSpingridApplication**(java.lang.String appdoc)

- **Usage**

- * Create the JSDLSpingridApplication, parsing a XML string.

5270

- **Parameters**

- * appdoc - The XML string to parse.

METHODS

- *getOperatingSystem*

5275 public JSDLSpingridOperatingSystem **getOperatingSystem**(
java.lang.String os)

- **Usage**

- * Search the list of supported operating systems, looking for a specific one, and return that one if it is found.

5280

- **Parameters**

- * os - The name of the OS to look for.

- **Returns** - The JSDLSpingridOperatingSystem if found, or null otherwise.

- *getOperatingSystems*

5285 public LinkedList **getOperatingSystems**()

- **Usage**

- * Get access to the actual list of operating systems.

- **Returns** - The actual list of operating systems.

5290

- *getSupportedOSes*

public int **getSupportedOSes**()

- **Usage**

- * Get an integer representing the support for OSes. Each bit represents support for one OS. See JSDL.OSNAMES[].

5295

- **Returns** - The OS support integer.

- *treeRepresentation*

public String treeRepresentation(int in)

- **Usage**

5300 * Return a multi-line string containing a tree representation of this
JSDLSpingridApplication.

- **Parameters**

 * in - The amount of indenting to apply.

- **Returns** - The tree representation.

5305

- *xmlRepresentation*

public String xmlRepresentation(int in)

- **Usage**

5310 * Return a multi-line string containing an XML representation of this
JSDLSpingridApplication.

- **Parameters**

 * in - The amount of indenting to apply.

- **Returns** - The XML representation.

CLASS JSDLSpingridDatasets

5315

This class represents the JSDLSpingridDatasets.

DECLARATION

```
public class JSDLSpingridDatasets
extends java.lang.Object
```

5320

CONSTRUCTORS

- *JSDLSpingridDatasets*

public JSDLSpingridDatasets(org.w3c.dom.Node datasets)

5325

- **Usage**

 * Create the JSDLSpingridDatasets, parsing a w3c dom Node.

- **Parameters**

 * datasets - The w3c dom Node to parse.

METHODS

- *getDataSetNames*

`public LinkedList getDataSetNames()`

- **Usage**

- * Get access to the actual list of dataset names.

- **Returns** - The actual list of dataset names of this JSDLSpingridDatasets.

- *treeRepresentation*

`public String treeRepresentation(int in)`

- **Usage**

- * Return a multi-line string containing a tree representation of this JSDLSpingridDatasets.

- **Parameters**

- * **in** - The amount of indenting to apply.

- **Returns** - The tree representation.

- *xmlRepresentation*

`public String xmlRepresentation(int in)`

- **Usage**

- * Return a multi-line string containing an XML representation of this JSDLSpingridDatasets.

- **Parameters**

- * **in** - The amount of indenting to apply.

- **Returns** - The XML representation.

CLASS JSDLSpingridExecutable

This class represents the JSDLSpingridExecutable.

DECLARATION

```
public class JSDLSpingridExecutable
    extends java.lang.Object
```

CONSTRUCTORS

- *JSDLSpingridExecutable*

`public JSDLSpingridExecutable(org.w3c.dom.Node execnode)`

- 5365 – **Usage**
 * Create the JSDLSpinggridExecutable, parsing a w3c dom Node.
- **Parameters**
 * **execnode** - The w3c dom Node to parse.

METHODS

- 5370 • *getExecutable*
 public String **getExecutable**()
- **Usage**
 * Get access to the executable.
- 5375 – **Returns** - The executable of this JSDLSpinggridExecutable.
-
- *getParameters*
 public LinkedList **getParameters**()
- **Usage**
 * Get access to the list of parameters.
- 5380 – **Returns** - The list of parameters of this JSDLSpinggridExecutable.
-
- *setExecutable*
 public void **setExecutable**(java.lang.String **exec**)
- 5385 – **Usage**
 * Set the executable of this JSDLSpinggridExecutable.
- **Parameters**
 * **exec** - The new executable.
-
- 5390 • *treeRepresentation*
 public String **treeRepresentation**(int **in**)
- **Usage**
 * Return a multi-line string containing a tree representation of this
 JSDLSpinggridExecutable.
- 5395 – **Parameters**
 * **in** - The amount of indenting to apply.
- **Returns** - The tree representation.
-
- *xmlRepresentation*
 public String **xmlRepresentation**(int **in**)
- 5400 – **Usage**
 * Return a multi-line string containing an XML representation of this
 JSDLSpinggridExecutable.
- **Parameters**
 * **in** - The amount of indenting to apply.
- 5405 – **Returns** - The XML representation.

CLASS JSDLSpingridOperatingSystem

This class represents the JSDLSpingridOperatingSystem.

5410 **DECLARATION**

```
public class JSDLSpingridOperatingSystem
extends java.lang.Object
```

CONSTRUCTORS

5415

- *JSDLSpingridOperatingSystem*
**public JSDLSpingridOperatingSystem(org.w3c.dom.Node
operatingsystems)**
 - **Usage**
 - * Create the JSDLSpingridOperatingSystem, parsing a w3c dom Node.
 - **Parameters**
 - * **operatingsystems** - The w3c dom Node to parse.

5420

METHODS

5425

- *getExecutables*
public LinkedList getExecutables()
 - **Usage**
 - * Get access to the exeuctables supported by this JSDLSpingridOperatingSystem.
 - **Returns** - The list of executables.
-
- *getOS*
public String getOS()
 - **Usage**
 - * Get access to the operating system name.
 - **Returns** - The operating system name of this JSDLSpingridOperatingSystem.
-
- *treeRepresentation*
public String treeRepresentation(int in)
 - **Usage**

5430

5435

5440

* Return a multi-line string containing a tree representation of this
JSDLSpinggridOperatingSystem.

– **Parameters**

* **in** - The amount of indenting to apply.

– **Returns** - The tree representation.

5445

• *xmlRepresentation*

`public String xmlRepresentation(int in)`

– **Usage**

* Return a multi-line string containing an XML representation of this
JSDLSpinggridOperatingSystem.

5450

– **Parameters**

* **in** - The amount of indenting to apply.

– **Returns** - The XML representation.

3.12 Package types

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
AgentSpecs	164
...no description...	
DownOrUploadStatus	165
...no description...	
Interval	166
...no description...	
PollRequest	167
...no description...	
PollResponse	169
...no description...	
ResourceProperties	171
...no description...	
<hr/>	

3.12.1 Classes

CLASS **AgentSpecs**

DECLARATION

```
public class AgentSpecs
extends java.lang.Object
```

CONSTRUCTORS

- *AgentSpecs*
public **AgentSpecs**()

METHODS

- *getCPUArchitecture*
public String **getCPUArchitecture**()
– **Returns** - A String with the CPU architecture
- *getCPUCount*
public int **getCPUCount**()
– **Returns** - An int with the number of CPU's
- *getCPUSpeed*
public long **getCPUSpeed**()
– **Returns** - A long with the CPU speed
- *getExclusiveExecution*
public boolean **getExclusiveExecution**()
– **Returns** - A Boolean if exclusive execution is allowed
- *getFileSystem_NORMAL_Capacity*
public long **getFileSystem_NORMAL_Capacity**()
– **Returns** - A long with the normal capacity
- *getFileSystem_NORMAL_MountPoint*
public String **getFileSystem_NORMAL_MountPoint**()

5505 – **Returns** - A String with the normal mount point

• *getFileSystem_TMP_Capacity*
 public long **getFileSystem_TMP_Capacity**()

 – **Returns** - A long with the TMP capacity

5510 • *getFileSystem_TMP_MountPoint*
 public String **getFileSystem_TMP_MountPoint**()

 – **Returns** - A String with the TMP mount point

5515 • *getNetworkBandwidth*
 public long **getNetworkBandwidth**()

 – **Returns** - A long with the bandwith of the network

 • *getOperatingSystem_Name*
 public String **getOperatingSystem_Name**()

5520 – **Returns** - A String with the name of the operating system

 • *getOperatingSystem_Version*
 public String **getOperatingSystem_Version**()

 – **Returns** - A String with the version of the operating system

5525 • *getPhysicalMemory*
 public long **getPhysicalMemory**()

 – **Returns** - A long with the physical memory

5530 • *getTrustedProjects*
 public LinkedList **getTrustedProjects**()

 – **Returns** - A LinkedList with trusted projects

5535 • *getVirtualMemory*
 public long **getVirtualMemory**()

 – **Returns** - A long with the virtual memory

 • *iterator*
 public Iterator **iterator**()

5540 – **Returns** -

CLASS **DownOrUploadStatus**

DECLARATION

5545 `public class DownOrUploadStatus`
`extends java.lang.Object`

FIELDS

- `public int status`
 —
- `public String failedMessage`
 —

CONSTRUCTORS

- *DownOrUploadStatus*
`public DownOrUploadStatus(int status, java.lang.String
 failedMessage)`

CLASS **Interval**

5560 DECLARATION

`public class Interval`
`extends java.lang.Object`

FIELDS

- 5565
- `public static final int SECSINDAY`
 —
 - `public static final String TIMEFORMATERR`
 —

5570 **CONSTRUCTORS**

- *Interval*
`public Interval(java.lang.String begin, java.lang.String end)`

METHODS

5575

- *CheckInterval*
`public void CheckInterval(int begin, int end)`

5580

- *getBegin*
`public int getBegin()`

- *getEnd*
`public int getEnd()`

CLASS PollRequest

5585

DECLARATION

<pre>public class PollRequest extends java.lang.Object</pre>

5590 **FIELDS**

- `public static final String REQUESTING_JOB`
 —
- `public static final String CALCULATING_JOB`
 —
- `public static final String IDLING`
 —
- `public static final String JOB_FINISHED`
 —
- `public static final String JOB_FAILED`
 —

5600

CONSTRUCTORS

- *PollRequest*

5605 `public PollRequest(java.lang.String state)`

- **Usage**

- * This method constructs the PollRequest type

- **Parameters**

- * **AState** - The state of the PollRequest

5610

METHODS

- *addKeyValue*

`public void addKeyValue(java.lang.String key, java.lang.String value)`

5615

- **Usage**

- * This methods add a key and a value

- **Parameters**

- * **key**, - mentioned key
 - * **value**, - mentioned value

5620

- *deleteValue*

`public void deleteValue(java.lang.String key)`

- **Usage**

- * deletes the key value pair with key

5625

- **Parameters**

- * **key**, - mentioned key

- *getIteratorOverDefaultFields*

`public static Iterator getIteratorOverDefaultFields()`

5630

- **Returns** - an iterator over the default fields

- *getState*

`public String getState()`

- **Usage**

5635

- * This method returns the state.

- **Returns** - The state of the PollRequest

- *getValue*

`public String getValue(java.lang.String key)`

5640

- **Usage**

* fetch a value associated with the key

– **Parameters**

* **key**, - mentioned key

– **Returns** - mentioned value

5645

• *iterator*

`public Iterator iterator()`

– **Returns** - an iterator over the keyValues list

CLASS **PollResponse**

5650

DECLARATION

```
public class PollResponse
extends java.lang.Object
```

5655

FIELDS

• `public static final String GIVE_JOB`

–

• `public static final String NO_JOB_AVAILABLE`

5660

–

• `public static final String OK`

–

• `public static final String STOP_CALCULATION`

–

5665

• `public static final String ERROR`

–

• `public static final String GIVING_ID`

–

CONSTRUCTORS

5670

- *PollResponse*

public **PollResponse**(java.lang.String action)

- **Usage**

- * Constructor

5675

- **Parameters**

- * action, - sets the action field

- *PollResponse*

public **PollResponse**(java.lang.String agentID, boolean
isSettingAgentID)

5680

- **Usage**

- * Constructor

- **Parameters**

- * agentID, - sets the agentID field

5685

- * isSettingAgentID, - must be true

- *PollResponse*

public **PollResponse**(java.lang.String action, java.lang.String jsdl)

- **Usage**

- * Constructor

5690

- **Parameters**

- * action, - set the action field

- * jsdl, - sets the jsdl field

METHODS

5695

- *getAgentID*

public String **getAgentID**()

- **Returns** - agentID field

- *getJSDL*

public String **getJSDL**()

5700

- **Returns** - jsdl field

- *getState*

public String **getState**()

5705

- **Returns** - action field

CLASS ResourceProperties

DECLARATION

5710

```
public class ResourceProperties
extends java.lang.Object
```

CONSTRUCTORS

5715

- *ResourceProperties*
public **ResourceProperties**()

METHODS

5720

- *getValuesOf*
public String **getValuesOf**(java.lang.String **property**)

3.13 Package configuration

Package Contents *Page*

Classes	
Options	173
<i>This class manages the configuration file</i>	

3.13.1 Classes

CLASS Options

This class manages the configuration file

DECLARATION

```
public class Options
extends java.lang.Object
```

CONSTRUCTORS

- *Options*
`public Options()`

METHODS

- *getAgentID*
`public static String getAgentID()`
– **Returns** - A String with the agent ID from the config file.

- *getDispatcherAddress*
`public static String getDispatcherAddress()`
– **Returns** - A String with the dispatcher address from the config file.

- *getHTTP*
`public static String getHTTP()`
– **Returns** - A String with the HTTP-port from the config file.

- *getProxy*
`public static String getProxy()`
– **Returns** - A String with the proxy address from the config file.

- *getProxyPort*
`public static String getProxyPort()`
– **Returns** - A String with the proxy port from the config file.

- *loadProperties*

5760 `public static void loadProperties()`

- **Usage**

- * Loads the properties from the config file into the variables. If no corresponding entry is found, the value of the variable remains the default value.

5765

- *setAgentID*

`public static void setAgentID(java.lang.String id)`

- **Parameters**

- * id - A String with the new agent ID to be saved.

5770

- *setDispatcherAddress*

`public static void setDispatcherAddress(java.lang.String address)`

- **Parameters**

- * address - A String with the new dispatcher address to be saved.

5775

- *setHTTP*

`public static void setHTTP(java.lang.String http)`

- **Parameters**

- * http - A String with the new HTTP-port to be saved.

5780

- *setProxy*

`public static void setProxy(java.lang.String address)`

- **Parameters**

- * address - A String with the new proxy address to be saved.

5785

- *setProxyPort*

`public static void setProxyPort(java.lang.String port)`

- **Parameters**

- * address - A String with the new proxy port to be saved.

5790

3.14 Package aTranslator

<i>Package Contents</i>	<i>Page</i>
-------------------------	-------------

Classes

ATranslator	176
--------------------------	-----

When a method in the Translator is called, it translates the parameters of the method into a GSP.

3.14.1 Classes

CLASS **ATranslator**

When a method in the Translator is called, it translates the parameters of the method into a GSP. The GSPs are then send to the Communicator. GRPs from the Communicator are evaluated so that the appropriate message can be send to the user through the Distributor.

DECLARATION

```
public class ATranslator
extends java.lang.Object
implements interfaces.IAResourceProviderCommands
```

CONSTRUCTORS

- *ATranslator*
`public ATranslator(java.lang.String AgentID, types.AgentSpecs specs)`

- **Usage**

- * Creates a new instance of the CCommunicator object

- **Parameters**

- * **AgentID** - The AgentID already present in the agent
 - * **specs** - The specification of the agent

METHODS

- *begin*
`public void begin()`

- **Usage**

- * Method that is called when an instance has been created and the interfaces have been set

- *sendListProjects*
`public String sendListProjects()`

- *sendPoll*
`public PollResponse sendPoll(types.PollRequest polling)`

- *setInterfaces*
`public void setInterfaces(interfaces.IACommunicator iaCommunicator)`

5830

- **Usage**
 - * Sets references to all interfaces this component might need
- **Parameters**
 - * `iaCommunicator` - The communicator interface

3.15 Package dTranslator

<i>Package Contents</i>	<i>Page</i>
-------------------------	-------------

5835

Classes	
DTranslator	179
<i>The Agent Translator has several functions: - Receive generic packages to the Agent Communicator.</i>	

3.15.1 Classes

CLASS DTranslator

5840 The Agent Translator has several functions: - Receive generic packages to the Agent Communicator.
- Send commands to the Agent Manager. - Translate commands to generic packages and generic packages to commands.

DECLARATION

5845

```
public class DTranslator
extends java.lang.Object
implements interfaces.IDPackageInput
```

CONSTRUCTORS

- 5850
 - *DTranslator*
`public DTranslator()`
 - Usage
 - * Creates a new instance of this class

METHODS

- 5855
 - *begin*
`public void begin()`
 - Usage
 - * Method that is called when an instance has been created and the interfaces have been set

- 5860
 - *PerformCommand*
`public GRP PerformCommand(genericpackages.GSP p)`

- 5865
 - *setInterfaces*
`public void setInterfaces(interfaces.IDResourceProviderCommands
idResourceProviderCommands, interfaces.IDUserCommands
idUserCommands, interfaces.IDProjectAdminCommands
idProjectAdminCommands, interfaces.IDSystemAdminCommands
5870 idSystemAdminCommands, interfaces.IDJobProviderCommands
idJobProviderCommands, interfaces.IDApplicationProviderCommands
idApplicationProviderCommands, interfaces.IDDataProviderCommands
idDataProviderCommands)`

– **Usage**

- 5875 * Sets the interfaces that this component might need

– **Parameters**

- * **idResourceProviderCommands** - The interface of the Resourceprovider
 commands
- 5880 * **idUserCommands** - The interface of the standard user commands
- * **idProjectAdminCommands** - The interface of the projectadmin commands
- * **idSystemAdminCommands** - The interface of the systemadmin commands
- * **idJobProviderCommands** - The interface of the jobprovider commands
- * **idApplicationProviderCommands** - The interface of the applicationprovider
 commands
- 5885 * **idDataProviderCommands** - The interface of the dataprovider commands

3.16 Package cLogger

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
CLogger	182
<i>The function of this component is to log the actions of the applications.</i>	
TimeStamper	183
<i>Class that returns timestamps of the current time.</i>	
<hr/>	

5890

3.16.1 Classes

CLASS **CLogger**

5895 The function of this component is to log the actions of the applications. It is capable of logging regular messages and error messages separately.

DECLARATION

```
public class CLogger
extends java.lang.Object
```

5900

CONSTRUCTORS

- *CLogger*
 public CLogger()

METHODS

5905

- *debug*
 public static void debug(java.lang.String msg)
 - **Usage**
 * Prints a message to standard output. Should be used only when the message is debug related.
 - **Parameters**
 * *msg* - The message to be printed

5910

-
- *error*
 public static void error(java.lang.String msg)
 - **Usage**
 * Writes a string to the error log file, prefixed with the current date and time
 - **Parameters**
 * *msg* - Message to write

5915

5920

-
- *error*
 public static void error(java.lang.String msg, boolean stop)
 - **Usage**

5925 * Writes a string to the error log file, prefixed with the current date and time.
 Additionally it checks the boolean whether or not to abort execution of the
 program.

– **Parameters**

5930 * **msg** - Message to write
 * **stop** - Whether or not the application should be terminated

• *exception*

`public static void exception(java.lang.Exception ex)`

– **Usage**

5935 * This method handles an exception, by printing the stacktrace to the screen.

– **Parameters**

 * **ex** - The exception to be handled.

• *exception*

5940 `public static void exception(java.lang.Exception ex, boolean stop)`

– **Usage**

 * This method handles an exception, by printing the stacktrace to the screen.
 Additionally, it stops the execution of the application if the boolean is set to
 true.

5945 – **Parameters**

 * **ex** - The exception to be handled.

• *log*

`public static void log(java.lang.String msg)`

5950 – **Usage**

 * Writes a string to the log file, prefixed with the current date and time

– **Parameters**

 * **msg** - String to write to the log file

CLASS **TimeStamper**

5955

Class that returns timestamps of the current time.

DECLARATION

<pre>public class TimeStamper extends java.lang.Object</pre>

5960

CONSTRUCTORS

- *TimeStamper*
`public TimeStamper()`

5965 METHODS

- *getDateStamp*
`public static String getDateStamp()`
 - **Usage**
 - * Returns the current date in the form yyyy/MM/dd
 - **Returns** - The current date as **String** in format yyyy/MM/dd

-
- *getDateTimeStamp*
`public static String getDateTimeStamp()`
 - **Usage**
 - * Returns the current date in the form yyyy/MM/dd HH:mm:ss
 - **Returns** - The current date and time as **String** in format yyyy/MM/dd HH:mm:ss

-
- *getTimeStamp*
`public static String getTimeStamp()`
 - **Usage**
 - * Returns the current time in the form HH:mm:ss
 - **Returns** - The current time as **String** in format HH:mm:ss

5985

3.17 Package aLogger

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
ALogger	186
<i>The function of this component is to log the actions of the applications.</i>	
TimeStamper	187
<i>Class that returns timestamps of the current time.</i>	

5990

3.17.1 Classes

CLASS **ALogger**

The function of this component is to log the actions of the applications. It is capable of logging
5995 regular messages and error messages separately.

DECLARATION

```
public class ALogger  
extends java.lang.Object
```

6000 CONSTRUCTORS

- *ALogger*
 public ALogger()

METHODS

6005

- *debug*
 public static void debug(java.lang.String msg)
 - **Usage**
 - * Prints a message to standard output. Should be used only when the message
6010 is debug related.
 - **Parameters**
 - * *msg* - The message to be printed

6015

- *error*
 public static void error(java.lang.String msg)
 - **Usage**
 - * Writes a string to the error log file, prefixed with the current date and time
 - **Parameters**
 - * *msg* - Message to write

6020

- *error*
 public static void error(java.lang.String msg, boolean stop)
 - **Usage**

6025 * Writes a string to the error log file, prefixed with the current date and time.
 Additionally it checks the boolean whether or not to abort execution of the
 program.

 – **Parameters**

 * **msg** - Message to write
 * **stop** - Whether or not the application should be terminated

6030

• *exception*
 public static void **exception**(java.lang.Exception **ex**)

 – **Usage**

 * This method handles an exception, by printing the stacktrace to the screen.

6035 – **Parameters**

 * **ex** - The exception to be handled.

• *exception*
 public static void **exception**(java.lang.Exception **ex**, boolean **stop**)

6040 – **Usage**

 * This method handles an exception, by printing the stacktrace to the screen.
 Additionally, it stops the execution of the application if the boolean is set to
 true.

 – **Parameters**

6045 * **ex** - The exception to be handled.

• *log*
 public static void **log**(java.lang.String **msg**)

 – **Usage**

6050 * Writes a string to the log file, prefixed with the current date and time

 – **Parameters**

 * **msg** - String to write to the log file

CLASS **TimeStamper**

6055 Class that returns timestamps of the current time.

DECLARATION

public class TimeStamper extends java.lang.Object

CONSTRUCTORS

- *TimeStamper*
`public TimeStamper()`

METHODS

- *getDateStamp*
`public static String getDateStamp()`
 - **Usage**
 - * Returns the current date in the form yyyy/MM/dd
 - **Returns** - The current date as **String** in format yyyy/MM/dd

- *getDateTimeStamp*
`public static String getDateTimeStamp()`
 - **Usage**
 - * Returns the current date in the form yyyy/MM/dd HH:mm:ss
 - **Returns** - The current date and time as **String** in format yyyy/MM/dd HH:mm:ss

- *getTimeStamp*
`public static String getTimeStamp()`
 - **Usage**
 - * Returns the current time in the form HH:mm:ss
 - **Returns** - The current time as **String** in format HH:mm:ss

3.18 Package aDataManager

Package Contents *Page*

6085

Interfaces

Upload 190
...no description...

Classes

ADataManager 190
This component will download data needed to perform jobs and will upload results when jobs are completed.

Download 192
...no description...

6090

FTPUpload 194
...no description...

HTTPUpload 197
...no description...

ReadLine 199
...no description...

6095

3.18.1 Interfaces

INTERFACE **Upload**

6100 DECLARATION

```
public interface Upload
```

METHODS

- *setRef*

```
public void setRef( types.DownOrUploadStatus ref )
```

- **Usage**

- * Sets reference to which should be set when finished

- **Parameters**

- * **ref**, - sets ref field in implementations

- *start*

```
public void start( )
```

- **Usage**

- * for runnables

3.18.2 Classes

CLASS **ADataManager**

6120 This component will download data needed to perform jobs and will upload results when jobs are completed.

DECLARATION

```
public class ADataManager  
  
extends java.lang.Object  
implements interfaces.IDataManager
```

FIELDS

- public static final int STATUS_SUCCESS

—

- 6130 • public static final int STATUS_FAILED

—

- public static final int STATUS_LOADING

—

CONSTRUCTORS

6135

- *ADataManager*

public **ADa**taManager()

— **Usage**

* Creates a new instance of the CCommunicator object

6140

METHODS

- *begin*

public void **begin**()

— **Usage**

6145 * Method that is called when an instance has been created and the interfaces
have been set

-
- *download*

6150 public DownOrUploadStatus **download**(java.lang.String **frompath**,
java.lang.String **topath**)

-
- *FTPUpload*

6155 public Upload **FTPUpload**(java.lang.String **frompath**, java.lang.String
topath)

-
- *HTTPUpload*

6160 public Upload **HTTPUpload**(java.lang.String **frompath**,
java.lang.String **topath**)

-
- *setInterfaces*

public void **setInterfaces**(interfaces.IAJobSchedulerForDataManager
iaJobScheduler)

— **Usage**


```

        * Sets references to all interfaces this component might need
6165    - Parameters
        * iaJobScheduler -
        _____
    • upload
      public DownOrUploadStatus upload( java.lang.String frompath,
6170      java.lang.String topath )

```

CLASS **Download**

DECLARATION

```

6175 public class Download
      extends java.lang.Thread

```

CONSTRUCTORS

```

    • Download
6180    public Download( java.lang.String frompath, java.lang.String topath
      )
    - Usage
      * Constructor
    - Parameters
6185    * frompath, - sets frompath field
      * topath, - sets topath field

```

METHODS

```

    • download
6190    public void download( java.lang.String frompath, java.lang.String
      topath )
    - Usage
      * This method will download a file from frompath to topath
    - Parameters
6195    * frompath, - mentioned frompath
      * topath, - mentioned topath
    _____
    • run
6200    public void run( )
    _____

```

- *setRef*
`public void setRef(types.DownOrUploadStatus ref)`
 - **Usage**
 - * Sets reference to which should be set when finished
 - **Parameters**
 - * `ref`, - sets ref

METHODS INHERITED FROM CLASS `java.lang.Thread`

- *activeCount*
6210 `public static int activeCount()`
- *checkAccess*
`public final void checkAccess()`
- 6215 • *countStackFrames*
`public native int countStackFrames()`
- *currentThread*
6220 `public static native Thread currentThread()`
- *destroy*
`public void destroy()`
- *dumpStack*
6225 `public static void dumpStack()`
- *enumerate*
`public static int enumerate(java.lang.Thread [])`
- 6230 • *getContextClassLoader*
`public ClassLoader getContextClassLoader()`
- *getName*
6235 `public final String getName()`
- *getPriority*
`public final int getPriority()`
- *getThreadGroup*
6240 `public final ThreadGroup getThreadGroup()`
- *holdsLock*
`public static native boolean holdsLock(java.lang.Object)`
- 6245 • *interrupt*
`public void interrupt()`
- *interrupted*
`public static boolean interrupted()`
- 6250 • *isAlive*
`public final native boolean isAlive()`
- *isDaemon*
6255 `public final boolean isDaemon()`
- *isInterrupted*
`public boolean isInterrupted()`
- 6260 • *join*
`public final void join()`

```

6265 • join
      public final synchronized void join( long    )
      • join
      public final synchronized void join( long    , int    )
      • resume
6270 public final void resume( )
      • run
      public void run( )
6275 • setContextClassLoader
      public void setContextClassLoader( java.lang.ClassLoader    )
      • setDaemon
6280 public final void setDaemon( boolean    )
      • setName
      public final void setName( java.lang.String    )
      • setPriority
6285 public final void setPriority( int    )
      • sleep
      public static native void sleep( long    )
6290 • sleep
      public static void sleep( long    , int    )
      • start
6295 public synchronized native void start( )
      • stop
      public final void stop( )
      • stop
6300 public final synchronized void stop( java.lang.Throwable    )
      • suspend
      public final void suspend( )
6305 • toString
      public String toString( )
      • yield
      public static native void yield( )

```

6310 CLASS **FTPUpload**

DECLARATION

```

public class FTPUpload
extends java.lang.Thread
implements Upload

```

6315

CONSTRUCTORS

- *FTPUpload*

public **FTPUpload**(java.lang.String **frompath**, java.lang.String
6320 **topath**)

- **Usage**

- * Constructor

- **Parameters**

- * **frompath**, - sets the frompath field

- * **topath**, - sets the topath field

METHODS

- *run*

public void **run**()

- *setRef*

public void **setRef**(types.DownOrUploadStatus **ref**)

- *upload*

public void **upload**()

- **Usage**

- * Upload file from frompath field to topath field

- **Exceptions**

- * java.io.IOException -

METHODS INHERITED FROM CLASS java.lang.Thread

- *activeCount*

public static int **activeCount**()

- *checkAccess*

public final void **checkAccess**()

- *countStackFrames*

public native int **countStackFrames**()

- *currentThread*

public static native Thread **currentThread**()

- *destroy*

public void **destroy**()

- *dumpStack*

public static void **dumpStack**()

- *enumerate*

public static int **enumerate**(java.lang.Thread [])

```

6365 • getContextClassLoader
      public ClassLoader getContextClassLoader( )
      • getName
        public final String getName( )
      • getPriority
6370 public final int getPriority( )
      • getThreadGroup
        public final ThreadGroup getThreadGroup( )
6375 • holdsLock
        public static native boolean holdsLock( java.lang.Object    )
      • interrupt
6380 public void interrupt( )
      • interrupted
        public static boolean interrupted( )
      • isAlive
6385 public final native boolean isAlive( )
      • isDaemon
        public final boolean isDaemon( )
6390 • isInterrupted
        public boolean isInterrupted( )
      • join
6395 public final void join( )
      • join
        public final synchronized void join( long    )
      • join
6400 public final synchronized void join( long , int    )
      • resume
        public final void resume( )
6405 • run
        public void run( )
      • setContextClassLoader
6410 public void setContextClassLoader( java.lang.ClassLoader    )
      • setDaemon
        public final void setDaemon( boolean    )
      • setName
6415 public final void setName( java.lang.String    )
      • setPriority
        public final void setPriority( int    )
6420 • sleep
        public static native void sleep( long    )
      • sleep
6425 public static void sleep( long , int    )
      • start
        public synchronized native void start( )
      • stop
6430 public final void stop( )
      • stop
        public final synchronized void stop( java.lang.Throwable    )

```

- 6435 • *suspend*
 public final void suspend()
- 6440 • *toString*
 public String toString()
- *yield*
 public static native void yield()

CLASS HTTPUpload

6445 DECLARATION

```
public class HTTPUpload
extends java.lang.Thread
implements Upload
```

CONSTRUCTORS

- 6450 • *HTTPUpload*
 public **HTTPUpload**(java.net.Socket *s*, java.net.URL *u*, java.io.File *f*)
- **Usage**
 * Constructor
- 6455 – **Parameters**
 * *s*, - sets sock field
 * *u*, - sets url field
 * *f*, - sets file field
- 6460

- *HTTPUpload*
 public **HTTPUpload**(java.lang.String *frompath*, java.lang.String *topath*)
- **Usage**
 * Constructor
- 6465 – **Parameters**
 * *frompath*, - path from which the file should be uploaded
 * *topath*, - path to which the file should be uploaded

METHODS

6470

- *run*

public void run()

- *setRef*

6475

public void setRef(types.DownOrUploadStatus ref)

METHODS INHERITED FROM CLASS java.lang.Thread

6480

- *activeCount*

public static int activeCount()

- *checkAccess*

public final void checkAccess()

- *countStackFrames*

6485

public native int countStackFrames()

- *currentThread*

public static native Thread currentThread()

6490

- *destroy*

public void destroy()

- *dumpStack*

6495

public static void dumpStack()

- *enumerate*

public static int enumerate(java.lang.Thread [])

- *getContextClassLoader*

6500

public ClassLoader getContextClassLoader()

- *getName*

public final String getName()

6505

- *getPriority*

public final int getPriority()

- *getThreadGroup*

6510

public final ThreadGroup getThreadGroup()

- *holdsLock*

public static native boolean holdsLock(java.lang.Object)

- *interrupt*

6515

public void interrupt()

- *interrupted*

public static boolean interrupted()

6520

- *isAlive*

public final native boolean isAlive()

- *isDaemon*

6525

public final boolean isDaemon()

- *isInterrupted*

public boolean isInterrupted()

- *join*

6530

public final void join()

- *join*
public final synchronized void join(long)
- 6535 • *join*
public final synchronized void join(long , int)
- *resume*
public final void resume()
- 6540 • *run*
public void run()
- *setContextClassLoader*
public void setContextClassLoader(java.lang.ClassLoader)
- 6545 • *setDaemon*
public final void setDaemon(boolean)
- 6550 • *setName*
public final void setName(java.lang.String)
- *setPriority*
public final void setPriority(int)
- 6555 • *sleep*
public static native void sleep(long)
- *sleep*
public static void sleep(long , int)
- 6560 • *start*
public synchronized native void start()
- 6565 • *stop*
public final void stop()
- *stop*
public final synchronized void stop(java.lang.Throwable)
- 6570 • *suspend*
public final void suspend()
- *toString*
public String toString()
- 6575 • *yield*
public static native void yield()

CLASS **ReadLine**

6580

DECLARATION

```
public class ReadLine
extends java.lang.Thread
```


6585 **CONSTRUCTORS**

• *ReadLine*

```
public ReadLine( java.io.BufferedReader i, aDataManager.HTTPUpload p
)
```

– **Usage**

* Read a line

– **Parameters**

* i, - the reader to read from

* p, - my parent

6595 **METHODS**

• *getRead*

```
public String getRead( )
```

– **Returns** - what is read

• *run*

```
public synchronized void run( )
```

METHODS INHERITED FROM CLASS java.lang.Thread

• *activeCount*

```
public static int activeCount( )
```

• *checkAccess*

```
public final void checkAccess( )
```

• *countStackFrames*

```
public native int countStackFrames( )
```

• *currentThread*

```
public static native Thread currentThread( )
```

• *destroy*

```
public void destroy( )
```

• *dumpStack*

```
public static void dumpStack( )
```

• *enumerate*

```
public static int enumerate( java.lang.Thread [] )
```

• *getContextClassLoader*

```
public ClassLoader getContextClassLoader( )
```

• *getName*

```
public final String getName( )
```

• *getPriority*

```
public final int getPriority( )
```

```

6635  • getThreadGroup
      public final ThreadGroup getThreadGroup( )
      • holdsLock
6640  public static native boolean holdsLock( java.lang.Object  )
      • interrupt
      public void interrupt( )
      • interrupted
6645  public static boolean interrupted( )
      • isAlive
      public final native boolean isAlive( )
6650  • isDaemon
      public final boolean isDaemon( )
      • isInterrupted
6655  public boolean isInterrupted( )
      • join
      public final void join( )
      • join
6660  public final synchronized void join( long  )
      • join
      public final synchronized void join( long  , int  )
6665  • resume
      public final void resume( )
      • run
      public void run( )
6670  • setContextClassLoader
      public void setContextClassLoader( java.lang.ClassLoader  )
      • setDaemon
6675  public final void setDaemon( boolean  )
      • setName
      public final void setName( java.lang.String  )
6680  • setPriority
      public final void setPriority( int  )
      • sleep
6685  public static native void sleep( long  )
      • sleep
      public static void sleep( long  , int  )
      • start
6690  public synchronized native void start( )
      • stop
      public final void stop( )
6695  • stop
      public final synchronized void stop( java.lang.Throwable  )
      • suspend
      public final void suspend( )
6700  • toString
      public String toString( )
      • yield
6705  public static native void yield( )

```

3.19 Package genericpackages

Package Contents	Page
------------------	------

Classes	
GPT type	203
<i>Type for use in the Generic Send Package</i>	
GRP	204
<i>Generic Receive Package: A generic package that is converted from a HTTP response and is formatted as two strings (status code and message).</i>	
GSP	205
<i>Generic Send Package: A generic package that can be converted to a HTTP request header and is formatted as a list of pairs of two strings (variable and value).</i>	

6710

3.19.1 Classes

CLASS **GPT**ype

Type for use in the Generic Send Package

DECLARATION

```
public class GPType
extends java.lang.Object
```

CONSTRUCTORS

- *GPT*ype

```
public GPType( java.lang.String  variable, java.lang.String  value )
```

- **Usage**

- * Constructor

- **Parameters**

- * **variable** -
 - * **value** -

METHODS

- *getValue*

```
public String getValue( )
```

- **Usage**

- * Returns the value corresponding to the variable

- **Returns** - String

- *getVariable*

```
public String getVariable( )
```

- **Usage**

- * Returns the variable type

- **Returns** - String

CLASS GRP

Generic Receive Package: A generic package that is converted from a HTTP response and is formatted as two strings (status code and message).

DECLARATION

```
public class GRP
extends java.lang.Object
```

CONSTRUCTORS

• *GRP*

```
public GRP( )
```

– **Usage**

- * Creates a new instance of this class
Constructor

METHODS

• *getMessage*

```
public String getMessage( )
```

– **Usage**

- * Returns the message of the GRP package

– **Returns** - String

• *getStatusCode*

```
public int getStatusCode( )
```

– **Usage**

- * Returns the statuscode of the GRP package

– **Returns** - int

• *getStatusMessage*

```
public String getStatusMessage( )
```

– **Usage**

- * Returns the statusmessage of the GRP package

– **Returns** - String

- *setMessage*

6780 `public void setMessage(java.lang.String message)`

- **Usage**

- * Sets the message of the GRP package

- **Parameters**

- * `message` -

- *setStatusCode*

6785 `public void setStatusCode(int statuscode)`

- **Usage**

- * Sets the statuscode of the GRP package

- **Parameters**

- * `statuscode` -

- *setStatusMessage*

6790 `public void setStatusMessage(java.lang.String statusmessage)`

- **Usage**

- * sets the statusmessage of the GRPType

- **Parameters**

- * `statusmessage` - String

CLASS GSP

6800 Generic Send Package: A generic package that can be converted to a HTTP request header and is formatted as a list of pairs of two strings (variable and value).

DECLARATION

6805 `public class GSP`

`extends java.util.Vector`

CONSTRUCTORS

- *GSP*

6810 `public GSP()`

- **Usage**

- * Constructor

METHODS

- 6815 • *add*
 `public void add(java.lang.String variable, java.lang.String value)`
 – **Usage**
 * Add the variable and the value to the package
 – **Parameters**
6820 * `variable` -
 * `value` -

- *contains*
 `public boolean contains(java.lang.String variable)`
6825 – **Usage**
 * * Check if the package contains the specific variable
 – **Parameters**
 * `variable` -
 – **Returns** - boolean
6830

- *getSize*
 `public int getSize()`
 – **Usage**
 * Get the number of elements in the package
6835 – **Returns** - int

- *getValueOf*
 `public String getValueOf(java.lang.String variable)`
 – **Usage**
6840 * Get the value corresponding to the variable
 – **Parameters**
 * `variable` -
 – **Returns** - String

- 6845 • *remove*
 `public void remove(java.lang.String variable)`
 – **Usage**
 * Remove the element corresponding to the variable
 – **Parameters**
6850 * `variable` -

METHODS INHERITED FROM CLASS `java.util.Vector`

- *add*
6855 public void add(int , java.lang.Object)
- *add*
 public synchronized boolean add(java.lang.Object)
- *addAll*
6860 public synchronized boolean addAll(java.util.Collection)
- *addAll*
 public synchronized boolean addAll(int , java.util.Collection)
- *addElement*
6865 public synchronized void addElement(java.lang.Object)
- *capacity*
6870 public synchronized int capacity()
- *clear*
 public void clear()
- *clone*
6875 public synchronized Object clone()
- *contains*
 public boolean contains(java.lang.Object)
- *containsAll*
6880 public synchronized boolean containsAll(java.util.Collection)
- *copyInto*
6885 public synchronized void copyInto(java.lang.Object [])
- *elementAt*
 public synchronized Object elementAt(int)
- *elements*
6890 public Enumeration elements()
- *ensureCapacity*
 public synchronized void ensureCapacity(int)
- *equals*
6895 public synchronized boolean equals(java.lang.Object)
- *firstElement*
6900 public synchronized Object firstElement()
- *get*
 public synchronized Object get(int)
- *hashCode*
6905 public synchronized int hashCode()
- *indexOf*
 public int indexOf(java.lang.Object)
- *indexOf*
6910 public synchronized int indexOf(java.lang.Object , int)
- *insertElementAt*
6915 public synchronized void insertElementAt(java.lang.Object , int)
- *isEmpty*
 public synchronized boolean isEmpty()

- *lastElement*
6920 public synchronized Object lastElement()
- *lastIndexOf*
public synchronized int lastIndexOf(java.lang.Object)
- 6925 • *lastIndexOf*
public synchronized int lastIndexOf(java.lang.Object , int)
- *remove*
6930 public synchronized Object remove(int)
- *remove*
public boolean remove(java.lang.Object)
- *removeAll*
6935 public synchronized boolean removeAll(java.util.Collection)
- *removeAllElements*
public synchronized void removeAllElements()
- 6940 • *removeElement*
public synchronized boolean removeElement(java.lang.Object)
- *removeElementAt*
6945 public synchronized void removeElementAt(int)
- *removeRange*
protected void removeRange(int , int)
- *retainAll*
6950 public synchronized boolean retainAll(java.util.Collection)
- *set*
public synchronized Object set(int , java.lang.Object)
- 6955 • *setElementAt*
public synchronized void setElementAt(java.lang.Object , int)
- *setSize*
6960 public synchronized void setSize(int)
- *size*
public synchronized int size()
- *subList*
6965 public synchronized List subList(int , int)
- *toArray*
public synchronized Object toArray()
- 6970 • *toArray*
public synchronized Object toArray(java.lang.Object [])
- *toString*
public synchronized String toString()
- 6975 • *trimToSize*
public synchronized void trimToSize()

METHODS INHERITED FROM CLASS `java.util.AbstractList`

6980 • *add*
 public void add(int , java.lang.Object)

6985 • *add*
 public boolean add(java.lang.Object)

6990 • *addAll*
 public boolean addAll(int , java.util.Collection)

6995 • *clear*
 public void clear()

7000 • *equals*
 public boolean equals(java.lang.Object)

7005 • *get*
 public abstract Object get(int)

7010 • *hashCode*
 public int hashCode()

7015 • *indexOf*
 public int indexOf(java.lang.Object)

7020 • *iterator*
 public Iterator iterator()

7025 • *lastIndexOf*
 public int lastIndexOf(java.lang.Object)

7030 • *listIterator*
 public ListIterator listIterator()

7035 • *listIterator*
 public ListIterator listIterator(int)

7040 • *remove*
 public Object remove(int)

7045 • *removeRange*
 protected void removeRange(int , int)

7050 • *set*
 public Object set(int , java.lang.Object)

7055 • *subList*
 public List subList(int , int)

METHODS INHERITED FROM CLASS `java.util.AbstractCollection`

7030 • *add*
 public boolean add(java.lang.Object)

7035 • *addAll*
 public boolean addAll(java.util.Collection)

7040 • *clear*
 public void clear()

7045 • *contains*
 public boolean contains(java.lang.Object)

- *containsAll*
public boolean containsAll(java.util.Collection)
- *isEmpty*
7045 public boolean isEmpty()
- *iterator*
public abstract Iterator iterator()
- *remove*
7050 public boolean remove(java.lang.Object)
- *removeAll*
7055 public boolean removeAll(java.util.Collection)
- *retainAll*
public boolean retainAll(java.util.Collection)
- *size*
7060 public abstract int size()
- *toArray*
public Object toArray()
- *toArray*
7065 public Object toArray(java.lang.Object [])
- *toString*
public String toString()

7070 3.20 Package aJobScheduler

Package Contents *Page*

Classes

AJobScheduler 212

This component is responsible for scheduling jobs.

3.20.1 Classes

7075 CLASS **AJobScheduler**

This component is responsible for scheduling jobs. This means that the component needs to: - poll for jobs. - download data through the DataManager which is needed to execute jobs. - execute jobs using the JobExecutor. - upload data through the DataManager when a job is finished executing. -
7080 signal the dispatcher when a job is finished executing.

DECLARATION

```
public class AJobScheduler
extends java.lang.Object
implements interfaces.IAJobSchedulerForDistributor,
interfaces.IAJobSchedulerForJobExecutor, interfaces.IAJobSchedulerForDataManager,
java.lang Runnable
```

7085 FIELDS

- public static final int MODE_SLEEPING
 - Constant for the mode SLEEPING
- public static final int MODE_REQUESTING
 - 7090 – Constant for the mode REQUESTING
- public static final int MODE_DOWNLOADING
 - Constant for the mode DOWNLOADING
- public static final int MODE_EXECUTING
 - Constant for the mode EXECUTING
- 7095 • public static final int MODE_UPLOADING
 - Constant for the mode UPLOADING
- public static final int MODE_JOBFINISHED
 - Constant for the status JOBFINISHED
- public static final int MODE_JOBFAILED
 - 7100 – Constant for the status JOBFAILED
- public static final int MODE_JOB CALCULATING
 - Constant for the status JOBCALCULATING

- `public static final int SECONDSBETWEENPOLLS`

- Time between poll requests are sent

- 7105
- `public int mode`

- The current mode of the agent

- `public int status`

- The current status of the agent

CONSTRUCTORS

7110

- *AJobScheduler*

`public AJobScheduler(types.AgentSpecs specs)`

- **Usage**

- * Creates a new instance of the CCommunicator object

7115

METHODS

- *AddInterval*

`public void AddInterval(types.Interval interval)`

7120

- *begin*

`public void begin()`

- **Usage**

- * Method that is called when an instance has been created and the interfaces have been set

7125

- *getCurrentmode*

`public int getCurrentmode()`

- *giveDownloadResult*

7130

`public void giveDownloadResult(int status)`

- *giveDownloadResult*

`public void giveDownloadResult(int status, java.lang.String detail)`

7135

- **Usage**

- * Callback function to inform the scheduler of the download result.

- **Parameters**

- * **status** - Status indicating success or failure.

- * **detail** - The download detail.

7140

- *giveResult*
public void giveResult(int status)
 - *giveUploadResult*
7145 public void giveUploadResult(int status)
 - *giveUploadResult*
public void giveUploadResult(int status, java.lang.String detail)
 - Usage
 - 7150 * Callback function to inform the scheduler of the upload result.
 - Parameters
 - * status - Status indicating success or failure.
 - * detail - The upload detail.
-
- 7155 • *receivePoll*
public void receivePoll(JSDL.JSDL job)
 - *RemoveInterval*
7160 public void RemoveInterval(types.Interval interval)
 - *run*
public void run()
 - *setInterfaces*
7165 public void setInterfaces(interfaces.IAJobExecutor iaJobExecutor,
interfaces.IADistributorForJobScheduler iaDistributor,
interfaces.IADataManager iaDataManager)
 - Usage
 - * Sets references to all interfaces this component might need
 - 7170 – Parameters
 - * iaJobExecutor -
 - * iaDistributor -
 - * iaDataManager -
-
- 7175 • *StartScheduling*
public void StartScheduling()
 - *StopJob*
7180 public void StopJob()
 - *StopScheduling*
public void StopScheduling()

3.21 Package exceptions

Package Contents

Page

Classes

	ApplicationException	216
7185	<i>...no description...</i>	
	ApplicationNotFoundException	217
	<i>...no description...</i>	
	ArgumentException	218
	<i>...no description...</i>	
7190	AuthenticationException	220
	<i>...no description...</i>	
	CommandArgumentException	222
	<i>Thrown when a argument on the command-line could not be parsed.</i>	
	CommandSyntaxInvalidException	223
7195	<i>Thrown when the syntax of the command-line arguments is invalid.</i>	
	CommunicationException	225
	<i>...no description...</i>	
	DataSetException	226
	<i>...no description...</i>	
7200	IntervalException	228
	<i>...no description...</i>	
	InvalidIdentifierException	230
	<i>Thrown while parsing an identifier which is invalid</i>	
	JobException	231
7205	<i>...no description...</i>	
	KeywordException	233
	<i>...no description...</i>	
	ProjectException	234
	<i>...no description...</i>	
7210	RoleException	236
	<i>...no description...</i>	
	SPINGRIDException	238
	<i>...no description...</i>	
	SyntaxInvalidException	239
7215	<i>...no description...</i>	
	TrustException	241
	<i>...no description...</i>	
	UserException	242
	<i>...no description...</i>	

7220

3.21.1 Classes

CLASS **ApplicationException**

DECLARATION

7225

```
public class ApplicationException
extends exceptions.SPINGRIDException
```

CONSTRUCTORS

7230

- *ApplicationException*
public ApplicationException()
 - **Usage**
 - * Constructs a new instance of this exception
-

7235

- *ApplicationException*
public ApplicationException(java.lang.String message)
 - **Usage**
 - * Constructs a new instance of this exception
-

7240

- *ApplicationException*
**public ApplicationException(java.lang.String message,
java.lang.Throwable cause)**
 - **Usage**
 - * Constructs a new instance of this exception
-

7245

- *ApplicationException*
public ApplicationException(java.lang.Throwable cause)
 - **Usage**
 - * Constructs a new instance of this exception

7250

METHODS INHERITED FROM CLASS **exceptions.SPINGRIDException**

(in 3.21.1, page 238)

METHODS INHERITED FROM CLASS **java.lang.Exception**

7255 METHODS INHERITED FROM CLASS `java.lang.Throwable`

- *fillInStackTrace*
public synchronized native Throwable fillInStackTrace()
- 7260 • *getCause*
public Throwable getCause()
- *getLocalizedMessage*
7265 public String getLocalizedMessage()
- *getMessage*
public String getMessage()
- *getStackTrace*
7270 public StackTraceElement getStackTrace()
- *initCause*
public synchronized Throwable initCause(java.lang.Throwable)
- 7275 • *printStackTrace*
public void printStackTrace()
- *printStackTrace*
7280 public void printStackTrace(java.io.PrintStream)
- *printStackTrace*
public void printStackTrace(java.io.PrintWriter)
- *setStackTrace*
7285 public void setStackTrace(java.lang.StackTraceElement [])
- *toString*
public String toString()

CLASS `ApplicationNotFoundException`

7290

DECLARATION

```
public class ApplicationNotFoundException
    extends java.lang.Exception
```

7295 CONSTRUCTORS

- *ApplicationNotFoundException*
public ApplicationNotFoundException()
 - **Usage**
 - * Constructs a new instance of this exception
- 7300 • *ApplicationNotFoundException*
public ApplicationNotFoundException(java.lang.String message)

– **Usage**

7305 * Constructs a new instance of this exception

• *ApplicationNotFoundException*

public **ApplicationNotFoundException**(java.lang.String message,
java.lang.Throwable cause)

7310 – **Usage**

 * Constructs a new instance of this exception

• *ApplicationNotFoundException*

public **ApplicationNotFoundException**(java.lang.Throwable cause)

7315 – **Usage**

 * Constructs a new instance of this exception

METHODS INHERITED FROM CLASS java.lang.Exception

METHODS INHERITED FROM CLASS java.lang.Throwable

7320

• *fillInStackTrace*

public synchronized native Throwable fillInStackTrace()

• *getCause*

7325 public Throwable getCause()

• *getLocalizedMessage*

public String getLocalizedMessage()

7330 • *getMessage*

public String getMessage()

• *getStackTrace*

7335 public StackTraceElement getStackTrace()

• *initCause*

public synchronized Throwable initCause(java.lang.Throwable)

• *printStackTrace*

7340 public void printStackTrace()

• *printStackTrace*

public void printStackTrace(java.io.PrintStream)

7345 • *printStackTrace*

public void printStackTrace(java.io.PrintWriter)

• *setStackTrace*

7350 public void setStackTrace(java.lang.StackTraceElement [])

• *toString*

public String toString()

CLASS **ArgumentException**

7355 **DECLARATION**

```
public class ArgumentException
extends exceptions.SPINGRIDException
```

CONSTRUCTORS

7360

- *ArgumentException*
 public **ArgumentException**()
 – **Usage**
 * Constructs a new instance of this exception

7365

- *ArgumentException*
 public **ArgumentException**(java.lang.String message)
 – **Usage**
 * Constructs a new instance of this exception

7370

- *ArgumentException*
 public **ArgumentException**(java.lang.String message,
 java.lang.Throwable cause)
 – **Usage**
 * Constructs a new instance of this exception

7375

- *ArgumentException*
 public **ArgumentException**(java.lang.Throwable cause)
 – **Usage**
 * Constructs a new instance of this exception

7380

METHODS INHERITED FROM CLASS exceptions.SPINGRIDException

(in 3.21.1, page 238)

METHODS INHERITED FROM CLASS java.lang.Exception

7385

METHODS INHERITED FROM CLASS `java.lang.Throwable`

- *fillInStackTrace*
7390 public synchronized native Throwable fillInStackTrace()
- *getCause*
 public Throwable getCause()
- *getLocalizedMessage*
7395 public String getLocalizedMessage()
- *getMessage*
 public String getMessage()
- *getStackTrace*
7400 public StackTraceElement getStackTrace()
- *initCause*
7405 public synchronized Throwable initCause(java.lang.Throwable)
- *printStackTrace*
 public void printStackTrace()
- *printStackTrace*
7410 public void printStackTrace(java.io.PrintStream)
- *printStackTrace*
 public void printStackTrace(java.io.PrintWriter)
- *setStackTrace*
7415 public void setStackTrace(java.lang.StackTraceElement [])
- *toString*
 public String toString()

7420 CLASS **AuthenticationException**

DECLARATION

```
public class AuthenticationException
extends exceptions.SPINGRIDException
```

7425

CONSTRUCTORS

- *AuthenticationException*
 public AuthenticationException()
- **Usage**
 * Constructs a new instance of this exception
- *AuthenticationException*
 public AuthenticationException(java.lang.String message)

7435 – **Usage**
 * Constructs a new instance of this exception

• *AuthenticationException*
 public **AuthenticationException**(java.lang.String message,
 7440 java.lang.Throwable cause)

 – **Usage**
 * Constructs a new instance of this exception

• *AuthenticationException*
 public **AuthenticationException**(java.lang.Throwable cause)
 7445

 – **Usage**
 * Constructs a new instance of this exception

METHODS INHERITED FROM CLASS `exceptions.SPINGRIDException`

7450 (in 3.21.1, page 238)

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

7455 • *fillInStackTrace*
 public synchronized native Throwable fillInStackTrace()

 • *getCause*
 public Throwable getCause()
 7460

 • *getLocalizedMessage*
 public String getLocalizedMessage()

 • *getMessage*
 public String getMessage()
 7465

 • *getStackTrace*
 public StackTraceElement getStackTrace()

7470 • *initCause*
 public synchronized Throwable initCause(java.lang.Throwable)

 • *printStackTrace*
 public void printStackTrace()
 7475

 • *printStackTrace*
 public void printStackTrace(java.io.PrintStream)

 • *printStackTrace*
 public void printStackTrace(java.io.PrintWriter)
 7480

 • *setStackTrace*
 public void setStackTrace(java.lang.StackTraceElement [])

7485 • *toString*
 public String toString()

CLASS `CommandArgumentException`

Thrown when a argument on the command-line could not be parsed.

7490 **DECLARATION**

```
public class CommandArgumentException
extends exceptions.SPINGRIDException
```

CONSTRUCTORS

7495

- *CommandArgumentException*
public `CommandArgumentException`()

- **Usage**

- * Creates an instance of the exception

7500

- *CommandArgumentException*
public `CommandArgumentException`(`java.lang.String` `message`)

- **Usage**

- * Creates an instance of the exception with a given message

7505

- **Parameters**

- * `message` -

- *CommandArgumentException*
**public `CommandArgumentException`(`java.lang.String` `message`,
`java.lang.Throwable` `cause`)**

7510

- **Usage**

- * Creates an instance of the exception with a message and a cause

- **Parameters**

- * `message` -

- * `cause` -

7515

- *CommandArgumentException*
public `CommandArgumentException`(`java.lang.Throwable` `cause`)

- **Usage**

- * Creates an instance of the exception with a cause

7520

- **Parameters**

- * `cause` -

METHODS INHERITED FROM CLASS `exceptions.SPINGRIDException`

7525 (in 3.21.1, page 238)

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

7530 • *fillInStackTrace*
 public synchronized native Throwable fillInStackTrace()

7535 • *getCause*
 public Throwable getCause()

 • *getLocalizedMessage*
 public String getLocalizedMessage()

7540 • *getMessage*
 public String getMessage()

 • *getStackTrace*
 public StackTraceElement getStackTrace()

7545 • *initCause*
 public synchronized Throwable initCause(java.lang.Throwable)

 • *printStackTrace*
 public void printStackTrace()

7550 • *printStackTrace*
 public void printStackTrace(java.io.PrintStream)

 • *printStackTrace*
 public void printStackTrace(java.io.PrintWriter)

7555 • *setStackTrace*
 public void setStackTrace(java.lang.StackTraceElement [])

7560 • *toString*
 public String toString()

CLASS **CommandSyntaxInvalidException**

Thrown when the syntax of the command-line arguments is invalid.

7565 DECLARATION

<pre>public class CommandSyntaxInvalidException extends exceptions.SPINGRIDException</pre>

CONSTRUCTORS

7570

- *CommandSyntaxInvalidException*
public CommandSyntaxInvalidException()

- **Usage**

- * Creates an instance of the exception

7575

- *CommandSyntaxInvalidException*
public CommandSyntaxInvalidException(java.lang.String message)

- **Usage**

- * Creates an instance of the exception with a given message

7580

- **Parameters**

- * **message** -

- *CommandSyntaxInvalidException*
**public CommandSyntaxInvalidException(java.lang.String message,
java.lang.Throwable cause)**

7585

- **Usage**

- * Creates an instance of the exception with a message and a cause

- **Parameters**

- * **message** -

- * **cause** -

7590

- *CommandSyntaxInvalidException*
public CommandSyntaxInvalidException(java.lang.Throwable cause)

- **Usage**

- * Creates an instance of the exception with a cause

7595

- **Parameters**

- * **cause** -

METHODS INHERITED FROM CLASS `exceptions.SPINGRIDException`

7600 (in 3.21.1, page 238)

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

- 7605 • *fillInStackTrace*
 public synchronized native Throwable fillInStackTrace()
- *getCause*
 public Throwable getCause()
- 7610 • *getLocalizedMessage*
 public String getLocalizedMessage()
- *getMessage*
 public String getMessage()
- 7615 • *getStackTrace*
 public StackTraceElement getStackTrace()
- 7620 • *initCause*
 public synchronized Throwable initCause(java.lang.Throwable)
- *printStackTrace*
 public void printStackTrace()
- 7625 • *printStackTrace*
 public void printStackTrace(java.io.PrintStream)
- *printStackTrace*
 public void printStackTrace(java.io.PrintWriter)
- 7630 • *setStackTrace*
 public void setStackTrace(java.lang.StackTraceElement [])
- 7635 • *toString*
 public String toString()

CLASS **CommunicationException**

DECLARATION

7640

public class CommunicationException
 extends java.lang.Exception

CONSTRUCTORS

- 7645 • *CommunicationException*
 public CommunicationException()
 – **Usage**
 * Constructs a new instance of this exception
- 7650 • *CommunicationException*
 public CommunicationException(java.lang.String message)

– **Usage**

* Constructs a new instance of this exception

• *CommunicationException*

public **CommunicationException**(java.lang.String message,
java.lang.Throwable cause)

– **Usage**

* Constructs a new instance of this exception

• *CommunicationException*

public **CommunicationException**(java.lang.Throwable cause)

– **Usage**

* Constructs a new instance of this exception

METHODS INHERITED FROM CLASS java.lang.Exception

METHODS INHERITED FROM CLASS java.lang.Throwable

• *fillInStackTrace*

public synchronized native Throwable **fillInStackTrace**()

• *getCause*

public Throwable **getCause**()

• *getLocalizedMessage*

public String **getLocalizedMessage**()

• *getMessage*

public String **getMessage**()

• *getStackTrace*

public StackTraceElement **getStackTrace**()

• *initCause*

public synchronized Throwable **initCause**(java.lang.Throwable)

• *printStackTrace*

public void **printStackTrace**()

• *printStackTrace*

public void **printStackTrace**(java.io.PrintStream)

• *printStackTrace*

public void **printStackTrace**(java.io.PrintWriter)

• *setStackTrace*

public void **setStackTrace**(java.lang.StackTraceElement [])

• *toString*

public String **toString**()

CLASS **DataSetException**

DECLARATION

```
public class DataSetException
7705 extends exceptions.SPINGRIDException
```

CONSTRUCTORS

- *DataSetException*

```
7710 public DataSetException( )
```

– **Usage**

* Constructs a new instance of this exception

- *DataSetException*

```
7715 public DataSetException( java.lang.String message )
```

– **Usage**

* Constructs a new instance of this exception

- *DataSetException*

```
7720 public DataSetException( java.lang.String message,
java.lang.Throwable cause )
```

– **Usage**

* Constructs a new instance of this exception

- *DataSetException*

```
7725 public DataSetException( java.lang.Throwable cause )
```

– **Usage**

* Constructs a new instance of this exception

METHODS INHERITED FROM CLASS exceptions.SPINGRIDException

(in 3.21.1, page 238)

METHODS INHERITED FROM CLASS java.lang.Exception

METHODS INHERITED FROM CLASS `java.lang.Throwable`

7735

- *fillInStackTrace*
`public synchronized native Throwable fillInStackTrace()`
- *getCause*
7740 `public Throwable getCause()`
- *getLocalizedMessage*
`public String getLocalizedMessage()`
- 7745 • *getMessage*
`public String getMessage()`
- *getStackTrace*
7750 `public StackTraceElement getStackTrace()`
- *initCause*
`public synchronized Throwable initCause(java.lang.Throwable)`
- *printStackTrace*
7755 `public void printStackTrace()`
- *printStackTrace*
`public void printStackTrace(java.io.PrintStream)`
- 7760 • *printStackTrace*
`public void printStackTrace(java.io.PrintWriter)`
- *setStackTrace*
7765 `public void setStackTrace(java.lang.StackTraceElement [])`
- *toString*
`public String toString()`

CLASS `IntervalException`

7770 DECLARATION

```
public class IntervalException
extends java.lang.Exception
```

CONSTRUCTORS

7775

- *IntervalException*
`public IntervalException()`
 - **Usage**
 - * Constructs a new instance of this exception
- 7780 • *IntervalException*
`public IntervalException(java.lang.String message)`

– **Usage**

* Constructs a new instance of this exception

7785

• *IntervalException*

**public IntervalException(java.lang.String message,
java.lang.Throwable cause)**

– **Usage**

* Constructs a new instance of this exception

7790

• *IntervalException*

public IntervalException(java.lang.Throwable cause)

– **Usage**

* Constructs a new instance of this exception

7795

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

7800

• *fillInStackTrace*

public synchronized native Throwable fillInStackTrace()

• *getCause*

public Throwable getCause()

7805

• *getLocalizedMessage*

public String getLocalizedMessage()

• *getMessage*

public String getMessage()

7810

• *getStackTrace*

public StackTraceElement getStackTrace()

7815

• *initCause*

public synchronized Throwable initCause(java.lang.Throwable)

• *printStackTrace*

public void printStackTrace()

7820

• *printStackTrace*

public void printStackTrace(java.io.PrintStream)

• *printStackTrace*

public void printStackTrace(java.io.PrintWriter)

7825

• *setStackTrace*

public void setStackTrace(java.lang.StackTraceElement [])

7830

• *toString*

public String toString()

CLASS InvalidIdentifierException

Thrown while parsing an identifier which is invalid

7835 **DECLARATION**

```
public class InvalidIdentifierException
extends exceptions.SPINGRIDException
```

CONSTRUCTORS

7840

- *InvalidIdentifierException*
public InvalidIdentifierException()

- **Usage**

- * Creates an instance of the exception

7845

- *InvalidIdentifierException*
public InvalidIdentifierException(java.lang.String message)

- **Usage**

- * Creates an instance of the exception with a given message

7850

- **Parameters**

- * **message** -

- *InvalidIdentifierException*
**public InvalidIdentifierException(java.lang.String message,
java.lang.Throwable cause)**

7855

- **Usage**

- * Creates an instance of the exception with a message and a cause

- **Parameters**

- * **message** -

- * **cause** -

7860

- *InvalidIdentifierException*
public InvalidIdentifierException(java.lang.Throwable cause)

- **Usage**

- * Creates an instance of the exception with a cause

7865

- **Parameters**

- * **cause** -

METHODS INHERITED FROM CLASS `exceptions.SPINGRIDException`

7870 (in 3.21.1, page 238)

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

7875 • *fillInStackTrace*
 public synchronized native Throwable fillInStackTrace()

7880 • *getCause*
 public Throwable getCause()

7885 • *getLocalizedMessage*
 public String getLocalizedMessage()

7885 • *getMessage*
 public String getMessage()

7890 • *getStackTrace*
 public StackTraceElement getStackTrace()

7895 • *initCause*
 public synchronized Throwable initCause(java.lang.Throwable)

7900 • *printStackTrace*
 public void printStackTrace()

7905 • *printStackTrace*
 public void printStackTrace(java.io.PrintStream)

7905 • *printStackTrace*
 public void printStackTrace(java.io.PrintWriter)

7905 • *setStackTrace*
 public void setStackTrace(java.lang.StackTraceElement [])

7905 • *toString*
 public String toString()

CLASS `JobException`

DECLARATION

7910 public class JobException
 extends exceptions.SPINGRIDException

CONSTRUCTORS

- 7915 • *JobException*
 public **JobException**()
 – **Usage**
 * Constructs a new instance of this exception

- 7920 • *JobException*
 public **JobException**(java.lang.String message)
 – **Usage**
 * Constructs a new instance of this exception

- 7925 • *JobException*
 public **JobException**(java.lang.String message, java.lang.Throwable
 cause)
 – **Usage**
 * Constructs a new instance of this exception

- 7930 • *JobException*
 public **JobException**(java.lang.Throwable cause)
 – **Usage**
 * Constructs a new instance of this exception

7935 **METHODS INHERITED FROM CLASS exceptions.SPINGRIDException**

(in 3.21.1, page 238)

METHODS INHERITED FROM CLASS java.lang.Exception

7940 **METHODS INHERITED FROM CLASS java.lang.Throwable**

- 7945 • *fillInStackTrace*
 public synchronized native Throwable **fillInStackTrace**()

- 7950 • *getCause*
 public Throwable **getCause**()

- 7950 • *getLocalizedMessage*
 public String **getLocalizedMessage**()

- 7950 • *getMessage*
 public String **getMessage**()

7955 • *getStackTrace*
 public StackTraceElement getStackTrace()

 • *initCause*
 public synchronized Throwable initCause(java.lang.Throwable)

7960 • *printStackTrace*
 public void printStackTrace()

 • *printStackTrace*
 public void printStackTrace(java.io.PrintStream)

7965 • *printStackTrace*
 public void printStackTrace(java.io.PrintWriter)

 • *setStackTrace*
 public void setStackTrace(java.lang.StackTraceElement [])

7970 • *toString*
 public String toString()

CLASS **KeywordException**

7975

DECLARATION

<pre>public class KeywordException extends exceptions.SPINGRIDException</pre>
--

7980

CONSTRUCTORS

 • *KeywordException*
 public **KeywordException**()

 – **Usage**
 * Constructs a new instance of this exception

7985 • *KeywordException*
 public **KeywordException**(java.lang.String message)

 – **Usage**
 * Constructs a new instance of this exception

7990 • *KeywordException*
 public **KeywordException**(java.lang.String message,
 java.lang.Throwable cause)

7995 – **Usage**
 * Constructs a new instance of this exception

- *KeywordException*
`public KeywordException(java.lang.Throwable cause)`

– **Usage**

* Constructs a new instance of this exception

METHODS INHERITED FROM CLASS `exceptions.SPINGRIDException`

(in 3.21.1, page 238)

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

- *fillInStackTrace*
`public synchronized native Throwable fillInStackTrace()`
- *getCause*
`public Throwable getCause()`
- *getLocalizedMessage*
`public String getLocalizedMessage()`
- *getMessage*
`public String getMessage()`
- *getStackTrace*
`public StackTraceElement getStackTrace()`
- *initCause*
`public synchronized Throwable initCause(java.lang.Throwable)`
- *printStackTrace*
`public void printStackTrace()`
- *printStackTrace*
`public void printStackTrace(java.io.PrintStream)`
- *printStackTrace*
`public void printStackTrace(java.io.PrintWriter)`
- *setStackTrace*
`public void setStackTrace(java.lang.StackTraceElement [])`
- *toString*
`public String toString()`

CLASS **ProjectException**

DECLARATION

```
public class ProjectException
8045 extends exceptions.SPINGRIDException
```

CONSTRUCTORS

- *ProjectException*

```
8050 public ProjectException( )
```

- **Usage**

* Constructs a new instance of this exception

- *ProjectException*

```
8055 public ProjectException( java.lang.String message )
```

- **Usage**

* Constructs a new instance of this exception

- *ProjectException*

```
8060 public ProjectException( java.lang.String message,
java.lang.Throwable cause )
```

- **Usage**

* Constructs a new instance of this exception

- *ProjectException*

```
8065 public ProjectException( java.lang.Throwable cause )
```

- **Usage**

* Constructs a new instance of this exception

METHODS INHERITED FROM CLASS exceptions.SPINGRIDException

8070 (in 3.21.1, page 238)

METHODS INHERITED FROM CLASS java.lang.Exception

METHODS INHERITED FROM CLASS `java.lang.Throwable`

8075

- *fillInStackTrace*
`public synchronized native Throwable fillInStackTrace()`
- *getCause*
8080 `public Throwable getCause()`
- *getLocalizedMessage*
`public String getLocalizedMessage()`
- 8085 • *getMessage*
`public String getMessage()`
- *getStackTrace*
8090 `public StackTraceElement getStackTrace()`
- *initCause*
`public synchronized Throwable initCause(java.lang.Throwable)`
- *printStackTrace*
8095 `public void printStackTrace()`
- *printStackTrace*
`public void printStackTrace(java.io.PrintStream)`
- 8100 • *printStackTrace*
`public void printStackTrace(java.io.PrintWriter)`
- *setStackTrace*
8105 `public void setStackTrace(java.lang.StackTraceElement [])`
- *toString*
`public String toString()`

CLASS **RoleException**

8110 DECLARATION

<pre>public class RoleException extends exceptions.SPINGRIDException</pre>

CONSTRUCTORS

8115

- *RoleException*
`public RoleException()`

 - **Usage**
 - * Constructs a new instance of this exception
- 8120 • *RoleException*
`public RoleException(java.lang.String message)`

– **Usage**

* Constructs a new instance of this exception

8125

• *RoleException*

public RoleException(java.lang.String message, java.lang.Throwable
cause)

– **Usage**

* Constructs a new instance of this exception

8130

• *RoleException*

public RoleException(java.lang.Throwable cause)

– **Usage**

* Constructs a new instance of this exception

8135

METHODS INHERITED FROM CLASS `exceptions.SPINGRIDException`

(in 3.21.1, page 238)

METHODS INHERITED FROM CLASS `java.lang.Exception`

8140

METHODS INHERITED FROM CLASS `java.lang.Throwable`

• *fillInStackTrace*

public synchronized native Throwable fillInStackTrace()

8145

• *getCause*

public Throwable getCause()

• *getLocalizedMessage*

public String getLocalizedMessage()

8150

• *getMessage*

public String getMessage()

• *getStackTrace*

public StackTraceElement getStackTrace()

8155

• *initCause*

public synchronized Throwable initCause(java.lang.Throwable)

8160

• *printStackTrace*

public void printStackTrace()

• *printStackTrace*

public void printStackTrace(java.io.PrintStream)

8165

• *printStackTrace*

public void printStackTrace(java.io.PrintWriter)

• *setStackTrace*

public void setStackTrace(java.lang.StackTraceElement [])

8170

• *toString*

public String toString()

8175 **CLASS SPINGRIDEException**

DECLARATION

```
public class SPINGRIDEException
extends java.lang.Exception
```

8180

CONSTRUCTORS

- *SPINGRIDEException*

```
public SPINGRIDEException( )
```

8185

– **Usage**

* Constructs a new instance of this exception

- *SPINGRIDEException*

```
public SPINGRIDEException( java.lang.String message )
```

8190

– **Usage**

* Constructs a new instance of this exception

- *SPINGRIDEException*

```
public SPINGRIDEException( java.lang.String message,
java.lang.Throwable cause )
```

8195

– **Usage**

* Constructs a new instance of this exception

- *SPINGRIDEException*

```
public SPINGRIDEException( java.lang.Throwable cause )
```

8200

– **Usage**

* Constructs a new instance of this exception

METHODS INHERITED FROM CLASS java.lang.Exception

8205 **METHODS INHERITED FROM CLASS java.lang.Throwable**

- *fillInStackTrace*

```
public synchronized native Throwable fillInStackTrace( )
```

8210 • *getCause*
 public Throwable getCause()

8215 • *getLocalizedMessage*
 public String getLocalizedMessage()

 • *getMessage*
 public String getMessage()

8220 • *getStackTrace*
 public StackTraceElement getStackTrace()

 • *initCause*
 public synchronized Throwable initCause(java.lang.Throwable)

8225 • *printStackTrace*
 public void printStackTrace()

 • *printStackTrace*
 public void printStackTrace(java.io.PrintStream)

8230 • *printStackTrace*
 public void printStackTrace(java.io.PrintWriter)

 • *setStackTrace*
 public void setStackTrace(java.lang.StackTraceElement [])

8235 • *toString*
 public String toString()

CLASS `SyntaxInvalidException`

8240

DECLARATION

```
public class SyntaxInvalidException
extends exceptions.SPINGRIDException
```

8245 CONSTRUCTORS

8250 • *SyntaxInvalidException*
 public **SyntaxInvalidException**()

 – Usage
 * Constructs a new instance of this exception

8255 • *SyntaxInvalidException*
 public **SyntaxInvalidException**(java.lang.String message)

 – Usage
 * Constructs a new instance of this exception

- *SyntaxInvalidException*

```
public SyntaxInvalidException( java.lang.String  message,
                               java.lang.Throwable  cause )
```

– **Usage**

* Constructs a new instance of this exception

- *SyntaxInvalidException*

```
public SyntaxInvalidException( java.lang.Throwable  cause )
```

– **Usage**

* Constructs a new instance of this exception

METHODS INHERITED FROM CLASS `exceptions.SPINGRIDException`

(in 3.21.1, page 238)

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

- *fillInStackTrace*

```
public synchronized native Throwable fillInStackTrace( )
```

- *getCause*

```
public Throwable getCause( )
```

- *getLocalizedMessage*

```
public String getLocalizedMessage( )
```

- *getMessage*

```
public String getMessage( )
```

- *getStackTrace*

```
public StackTraceElement getStackTrace( )
```

- *initCause*

```
public synchronized Throwable initCause( java.lang.Throwable  )
```

- *printStackTrace*

```
public void printStackTrace( )
```

- *printStackTrace*

```
public void printStackTrace( java.io.PrintStream  )
```

- *printStackTrace*

```
public void printStackTrace( java.io.PrintWriter  )
```

- *setStackTrace*

```
public void setStackTrace( java.lang.StackTraceElement []  )
```

- *toString*

```
public String toString( )
```

CLASS **TrustException**

DECLARATION

```

8310 public class TrustException
      extends exceptions.SPINGRIDException

```

CONSTRUCTORS

- 8315 • *TrustException*
 public TrustException()

 – **Usage**
 * Constructs a new instance of this exception

- 8320 • *TrustException*
 public TrustException(java.lang.String message)

 – **Usage**
 * Constructs a new instance of this exception

- 8325 • *TrustException*
 public TrustException(java.lang.String message, java.lang.Throwable cause)

 – **Usage**
 * Constructs a new instance of this exception

- 8330 • *TrustException*
 public TrustException(java.lang.Throwable cause)

 – **Usage**
 * Constructs a new instance of this exception

METHODS INHERITED FROM CLASS **exceptions.SPINGRIDException**

8335 (in 3.21.1, page 238)

METHODS INHERITED FROM CLASS **java.lang.Exception**

METHODS INHERITED FROM CLASS `java.lang.Throwable`

8340

- *fillInStackTrace*
`public synchronized native Throwable fillInStackTrace()`
- *getCause*
8345 `public Throwable getCause()`
- *getLocalizedMessage*
`public String getLocalizedMessage()`
- 8350 • *getMessage*
`public String getMessage()`
- *getStackTrace*
8355 `public StackTraceElement getStackTrace()`
- *initCause*
`public synchronized Throwable initCause(java.lang.Throwable)`
- *printStackTrace*
8360 `public void printStackTrace()`
- *printStackTrace*
`public void printStackTrace(java.io.PrintStream)`
- 8365 • *printStackTrace*
`public void printStackTrace(java.io.PrintWriter)`
- *setStackTrace*
8370 `public void setStackTrace(java.lang.StackTraceElement [])`
- *toString*
`public String toString()`

CLASS `UserException`

8375 DECLARATION

```
public class UserException
extends exceptions.SPINGRIDException
```

CONSTRUCTORS

8380

- *UserException*
`public UserException()`
 - **Usage**
 - * Constructs a new instance of this exception
- 8385 • *UserException*
`public UserException(java.lang.String message)`

– **Usage**

* Constructs a new instance of this exception

• *UserException*

public UserException(java.lang.String message, java.lang.Throwable cause)

– **Usage**

* Constructs a new instance of this exception

• *UserException*

public UserException(java.lang.Throwable cause)

– **Usage**

* Constructs a new instance of this exception

METHODS INHERITED FROM CLASS `exceptions.SPINGRIDException`

(in 3.21.1, page 238)

METHODS INHERITED FROM CLASS `java.lang.Exception`

METHODS INHERITED FROM CLASS `java.lang.Throwable`

• *fillInStackTrace*

public synchronized native Throwable fillInStackTrace()

• *getCause*

public Throwable getCause()

• *getLocalizedMessage*

public String getLocalizedMessage()

• *getMessage*

public String getMessage()

• *getStackTrace*

public StackTraceElement getStackTrace()

• *initCause*

public synchronized Throwable initCause(java.lang.Throwable)

• *printStackTrace*

public void printStackTrace()

• *printStackTrace*

public void printStackTrace(java.io.PrintStream)

• *printStackTrace*

public void printStackTrace(java.io.PrintWriter)

• *setStackTrace*

public void setStackTrace(java.lang.StackTraceElement [])

• *toString*

public String toString()

8440

3.22 Package cCommunicator

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
CCommunicator	245
<i>This component has the function to communicate with the dispatcher.</i>	
<hr/>	

3.22.1 Classes

CLASS CCommunicator

This component has the function to communicate with the dispatcher.

DECLARATION

```
public class CCommunicator
extends java.lang.Object
implements interfaces.ICCommunicator
```

CONSTRUCTORS

- *CCommunicator*
public CCommunicator()
 - Usage
 - * Creates a new instance of this class

METHODS

- *begin*
public void begin()
 - Usage
 - * Method that is called when an instance has been created and the interfaces have been set
- *send*
public GRP send(genericpackages.GSP pack, java.net.InetAddress dispatchAddr)
- *setInterfaces*
public void setInterfaces()
 - Usage
 - * Sets references to all interfaces this component might need

3.23 Package HTTPMessage

Package Contents *Page*

8475

Classes

HTTPRequest 247

*The HTTPRequest is a general abstraction of the HTTP protocol: - Easy
insertion of attributes - Parsing existing requests - Uses HTTP 1/1*

HTTPResponse 249

*The HTTPResponse is a general abstraction of the HTTP protocol: - Easy
insertion of attributes - Parsing existing responses - Uses HTTP 1/1*

KeyValues 251

...no description...

8480

3.23.1 Classes

CLASS `HttpRequest`

8485 The `HttpRequest` is a general abstraction of the HTTP protocol: - Easy insertion of attributes -
 Parsing existing requests - Uses HTTP 1/1

DECLARATION

```
public class HttpRequest
extends java.lang.Object
```

8490

METHODS

- *addValue*

```
public void addValue( java.lang.String key, java.lang.String value )
```

8495

- Usage

- * Adds an attribute with key and value Or adds a value to an attribute with key

- Parameters

- * `key` - Mentioned key
 - * `value` - Mentioned value

8500

- *addVariable*

```
public void addVariable( java.lang.String key, java.lang.String value
)
```

8505

- Usage

- * Adds a variable with key and value

- Parameters

- * `key` - Mentioned key
 - * `value` - Mentioned value

8510

- *build*

```
public static HttpRequest build( java.lang.String HTTPVersion,
java.lang.String method, java.lang.String URL, java.lang.String
content )
```

8515

- Usage

- * Constructor-like

- Parameters

- * `HTTPVersion` - Sets HTTP version field

8520 * **method** - Sets method field
 * **URL** - Sets URL field
 * **keysAndValues** - Sets keysAndValues field
 * **contentVariables** - Sets contentVariables version field
 – **Returns** - A HTTPRequest with data conforming the parameters

8525 • *getAValue*
 public String **getAValue**(java.lang.String key)
 – **Usage**
 * Get a value from a certain attribute with key
 – **Parameters**
 * **key** - Is the key to get a value of
 – **Returns** - A found value

8530 • *getBytes*
 public byte **getBytes**()
 – **Usage**
 * Converts packet to sequence of bytes
 – **Returns** - this packet as array of bytes

8540 • *getHTTPVersion*
 public String **getHTTPVersion**()
 – **Returns** - HTTP Version field

8545 • *getMethod*
 public String **getMethod**()
 – **Returns** - method field

8550 • *getURL*
 public String **getURL**()
 – **Returns** - URL field

8555 • *getValues*
 public String **getValues**(java.lang.String key)
 – **Usage**
 * Get values from a certain attribute with key
 – **Parameters**
 * **key** - Is the key to get the values of
 – **Returns** - The found values

8560 • *getVariable*
 public String **getVariable**(java.lang.String key)

- **Usage**
 - * Get the value from a variable with key

- **Parameters**

- * **key** - Is the key to get the value of

- **Returns** - Found value

- *isCorrect*

```
public boolean isCorrect( )
```

- **Returns** - Correctness of the HTTP message

- *iterator*

```
public Iterator iterator( )
```

- *parse*

```
public static HTTPRequest parse( java.io.BufferedReader is )
```

- **Usage**

- * Parsed data to HTTPRequest

- **Parameters**

- * **is** - Is the input stream, in the form of a buffered reader

- **Returns** - A HTTPRequest with data conforming the parsed is

- *toString*

```
public String toString( )
```

- **Usage**

- * Converts packet to string, in order to be logged

- **Returns** - this packet as string

CLASS HTTPResponse

The HTTPResponse is a general abstraction of the HTTP protocol: - Easy insertion of attributes -
Parsing existing responses - Uses HTTP 1/1

DECLARATION

```
public class HTTPResponse  
extends java.lang.Object
```

METHODS

• *addValue*

```
public void addValue( java.lang.String  key, java.lang.String  value )
```

– **Usage**

* Adds an attribute with key and value Or adds a value to an attribute with key

– **Parameters**

* **key** - Mentioned key
* **value** - Mentioned value

• *build*

```
public static HTTPResponse build( java.lang.String  HTTPVersion, int  
codeN, java.lang.String  codeS, java.lang.String  content )
```

– **Usage**

* Constructor-like

– **Parameters**

* **HTTPVersion** - Sets the HTTP version field
* **codeN** - Sets the statuscode number field
* **codeS** - Sets the status message field
* **content** - Sets the content field

– **Returns** - A HTTPResponse with data conforming the parameters

• *getAValue*

```
public String getAValue( java.lang.String  key )
```

– **Usage**

* Get a value from a certain attribute with key

– **Parameters**

* **key** - Is the key to get a value of

– **Returns** - A found value

• *getBytes*

```
public byte getBytes( )
```

– **Usage**

* Converts packet to sequence of bytes

– **Returns** - this packet as array of bytes

• *getCodeN*

```
public int getCodeN( )
```

– **Returns** - statuscode number field

• *getCodeS*

```
public String getCodeS( )
```

– **Returns** - status message field

- *getContent*

public String **getContent**()

– **Returns** - content field

- *getHTTPVersion*

public String **getHTTPVersion**()

– **Returns** - HTTP Version field

- *getValues*

public String **getValues**(java.lang.String key)

– **Usage**

* Get values from a certain attribute with key

– **Parameters**

* **key** - Is the key to get the values of

– **Returns** - The found values

- *isCorrect*

public boolean **isCorrect**()

– **Returns** - Correctness of the HTTP message

- *parse*

public static HTTPResponse **parse**(java.io.BufferedReader is)

– **Usage**

* Parsed data to HTTPResponse

– **Parameters**

* **is** - Is the input stream, in the form of a buffered reader

– **Returns** - A HTTPResponse with data conforming the parsed is

- *toString*

public String **toString**()

– **Usage**

* Converts packet to string, in order to be logged

– **Returns** - this packet as string

CLASS KeyValues

8675 **DECLARATION**

```
public class KeyValues
extends java.lang.Object
```

FIELDS

8680

- public String key
 - Key
- public String values
 - Associated values

8685 **CONSTRUCTORS**

- *KeyValues*

```
public KeyValues( java.lang.String  key, java.lang.String [] values )
```

 - **Usage**
 - * Is an attribute or variable, with a key and one or more values
 - **Parameters**
 - * **key** - Key
 - * **values** - Associated values

8690

3.24 Package cIOProcessor

Package Contents

Page

8695

Classes

CIOProcessor 254

This component has the function to process inputs from the outside world and process outputs from the other components of the client.

CommandArgument 255

This class holds a commandargument of the type “<type>”: “<param>” where “<type>” is the type of the argument and “<param>” is the parameter.

CommandErrors 257

Class containing all error messages and descriptions that can be outputted by the CommandLineParser

CommandParser 261

This class parses the an array of arguments.

8700

3.24.1 Classes

CLASS **CIOProcessor**

This component has the function to process inputs from the outside world and process outputs from the other components of the client.

DECLARATION

```
public class CIOProcessor
extends java.lang.Object
implements interfaces.ICInputProcessor, interfaces.ICOutputProcessor
```

CONSTRUCTORS

- *CIOProcessor*
public CIOProcessor()
 - **Usage**
 - * Constructs a new instance of this class

METHODS

- *begin*
public void begin()
 - **Usage**
 - * This method is called right after the component is created and the interfaces are set.

-
- *Input*
public void Input(java.lang.String [] args)

-
- *Output*
public void Output(java.lang.String str)

-
- *setInterfaces*
**public void setInterfaces(interfaces.ICUserCommands icUserCommands,
interfaces.ICProjectAdminCommands icProjectAdminCommands,
interfaces.ICSystemAdminCommands icSystemAdminCommands,
interfaces.ICJobProviderCommands icJobProviderCommands,
interfaces.ICApplicationProviderCommands
icApplicationProviderCommands, interfaces.ICDataProviderCommands
icDataProviderCommands)**

- **Usage**

- * Sets all interfaces this component may use.

- **Parameters**

- * `icUserCommands` -
 - * `icProjectAdminCommands` -
 - * `icSystemAdminCommands` -
 - * `icJobProviderCommands` -
 - * `icApplicationProviderCommands` -
 - * `icDataProviderCommands` -

CLASS `CommandArgument`

This class holds a commandargument of the type “<type>”:“<param>”where “<type>”is the type of the argument and “<param>”is the parameter.

DECLARATION

```
public class CommandArgument
    extends java.lang.Object
```

FIELDS

- `public static final String ROLE_APPPROV`
 - Constant for the paramter 'application provider' in a `CommandArgument` of type role.
- `public static final String ROLE_DATPROV`
 - Constant for the paramter 'data provider' in a `CommandArgument` of type role.
- `public static final String ROLE_JOBPROV`
 - Constant for the paramter 'job provider' in a `CommandArgument` of type role.
- `public static final String ROLE_PROJADMIN`
 - Constant for the paramter 'project admin' in a `CommandArgument` of type role.
- `public static final String ROLE_SYSADMIN`
 - Constant for the paramter 'system admin' in a `CommandArgument` of type role.
- `public static final int TYPE_APPLICATION`
 - A type constant which defines a `CommandArgument` as an application argument
- `public static final int TYPE_DATA`

- A type constant which defines a **CommandArgument** as a data argument
- 8775 • **public static final int TYPE_FILE**
 - A type constant which defines a **CommandArgument** as a file argument
- **public static final int TYPE_JOB**
 - A type constant which defines a **CommandArgument** as a job argument
- **public static final int TYPE_PROJECT**
 - 8780 – A type constant which defines a **CommandArgument** as a project argument
- **public static final int TYPE_RESOURCE**
 - A type constant which defines a **CommandArgument** as a resource argument
- **public static final int TYPE_ROLE**
 - A type constant which defines a **CommandArgument** as a role argument
- 8785 • **public static final int TYPE_URL**
 - A type constant which defines a **CommandArgument** as a URL argument
- **public static final int TYPE_USER**
 - A type constant which defines a **CommandArgument** as a user argument

CONSTRUCTORS

- 8790 • *CommandArgument*

```
public CommandArgument( java.lang.String  arg )
```

 - **Usage**
 - * Constructs a new **CommandArgument** from a given string. It first checks if the
 - 8795 string has the right form and if so the new object is constructed
 - **Parameters**
 - * The - string to be 'put in' the **CommandArgument**

METHODS

- 8800 • *getParam*

```
public String getParam( )
```

 - **Usage**
 - * Returns the parameter of the **CommandArgument**.
 - **Returns** - The parameter of the **CommandArgument**
- 8805 • *getType*

```
public int getType( )
```

- **Usage**
 - * Returns the type of the `CommandArgument`.
- **Returns** - The type of the `CommandArgument`

- *isValidArgument*

```
public static boolean isValidArgument( java.lang.String arg )
```

- **Usage**
 - * Checks if a string has the right form of an argument. The string should be at least 5 characters long and have a ':' as a fourth character.
- **Parameters**
 - * `arg` -
- **Returns** -

- *isValidRole*

```
public boolean isValidRole( )
```

- **Usage**
 - * This method checks whether the `CommandArgument` is of type role and if the parameter is a valid role constant.
- **Returns** - Whether or not the `CommandArgument` contains a valid role.

CLASS `CommandErrors`

Class containing all error messages and descriptions that can be outputted by the `CommandLineParser`

DECLARATION

```
public class CommandErrors
    extends java.lang.Object
```

FIELDS

- `public static final String ERR_AMBIGIOUS`
 - An error message indicating that the command is ambiguous.
- `public static final String ERR_ASROLE`
 - An error message indicating that the `as <role>construct` is invalid.
- `public static final String ERR_COMMANDS`

- An error message that should be put when an invalid command was entered.
- public static final String ERR_FILE_NOT_FOUND
 - An error message that indicates that a file could not be found.
- 8845 • public static final String ERR_ROLE_INVALID
 - An error indicating that one of the roles in a command is invalid.
- public static final String ERR_URL_INVALID
 - An error indicating that one of the urls in a command is invalid.
- public static final String ERR_HELP_ADD_APP
 - 8850 – The description belonging to the command "add app"
- public static final String ERR_HELP_ADD_DATA
 - The description belonging to the command "add data"
- public static final String ERR_HELP_ADD_JOB
 - The description belonging to the command "add job"
- 8855 • public static final String ERR_HELP_ADD_PROJECT
 - The description belonging to the command "add project"
- public static final String ERR_HELP_ADD_USER
 - The description belonging to the command "add user"
- public static final String ERR_HELP_ASSIGN
 - 8860 – The description belonging to the command "assign"
- public static final String ERR_HELP_DISTRICT
 - The description belonging to the command "distrust"
- public static final String ERR_HELP_HELP
 - The description belonging to the command "help"
- 8865 • public static final String ERR_HELP_INFO
 - The description belonging to the command "info"
- public static final String ERR_HELP_LIST_APPS
 - The description belonging to the command "list apps"
- public static final String ERR_HELP_LIST_DATA
 - 8870 – The description belonging to the command "list data"
- public static final String ERR_HELP_LIST_JOBS
 - The description belonging to the command "list jobs"

- public static final String ERR_HELP_LIST_PROJECTS
 - The description belonging to the command "list projects"
- 8875 • public static final String ERR_HELP_LIST_USERS
 - The description belonging to the command "list users"
- public static final String ERR_HELP_PASSWD
 - The description belonging to the command "passwd"
- 8880 • public static final String ERR_HELP_REMOVE
 - The description belonging to the command "remove"
- public static final String ERR_HELP_TRUST
 - The description belonging to the command "trust"
- public static final String ERR_HELP_UNASSIGN
 - The description belonging to the command "unassign"
- 8885 • public static final String ERR_SYNTAX_ADD_APP
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_ADD_DATA
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- 8890 • public static final String ERR_SYNTAX_ADD_JOB
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_ADD_PROJECT
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- 8895 • public static final String ERR_SYNTAX_ADD_USER
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- 8900 • public static final String ERR_SYNTAX_ASSIGN
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_DISTRUST
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- 8905 • public static final String ERR_SYNTAX_HELP

- An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_INFO
 - 8910 – An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_LIST_APPS
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- 8915 • public static final String ERR_SYNTAX_LIST_DATA
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_LIST_JOBS
 - 8920 – An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_LIST_PROJECTS
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_LIST_USERS
 - 8925 – An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_PASSWD
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- 8930 • public static final String ERR_SYNTAX_REMOVE
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_TRUST
 - 8935 – An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_SYNTAX_UNASSIGN
 - An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_WRONG_ROLE
 - 8940 – An error message indicating the syntax for the command was incorrect and gives the correct usage.
- public static final String ERR_WRONG_TYPE
 - An error message indicating that one of the arguments in the command is of the wrong type

CONSTRUCTORS

- *CommandErrors*
`public CommandErrors()`

METHODS

- *getArgumentExpected*
`public static String getArgumentExpected(java.lang.String syntax)`
 - **Usage**
 - * Returns an error message indicating that there is at least one argument missing at the end of the line and the syntax of a specified command
 - **Parameters**
 - * `syntax` - The syntax
 - **Returns** - String with the error message
- *getInvalidSyntaxError*
`public static String getInvalidSyntaxError(java.lang.String token, java.lang.String syntax)`
 - **Usage**
 - * Returns an error message that parsing failed on a specific token.
 - **Parameters**
 - * `token` - The token onw hich parsing failed
 - * `syntax` - The correct syntax of the command on which parsing failed
 - **Returns** - The error message
- *getTooManyArgumentsError*
`public static String getTooManyArgumentsError(java.lang.String syntax)`
 - **Usage**
 - * Returns an error message indicating that there is at least one argument too much at the end of the line and the syntax of a specified command
 - **Parameters**
 - * `syntax` - The syntax
 - **Returns** - String with the error message

CLASS **CommandParser**

This class parses the an array of arguments. If parsing fails an exception is thrown If it succeeds the command is interpreted and corresponding action is taken.

DECLARATION

```
public class CommandParser
extends java.lang.Object
```

CONSTRUCTORS

• *CommandParser*

```
public CommandParser( interfaces.IApplicationProviderCommands
appcommands, interfaces.IDataProviderCommands datcommands,
interfaces.IJobProviderCommands jobcommands,
interfaces.IProjectAdminCommands projcommands,
interfaces.ICSystemAdminCommands syscommands,
interfaces.IUserCommands usercommands, java.lang.String [] args )
```

– **Usage**

* Constructs a new instance of the CommandParser

– **Parameters**

* **args** - The sentence (words) on which the CommandParser will be working.

* **adistributor** - The interface with methods which the parser may call.

METHODS

• *getArguments*

```
public String getArguments( )
```

– **Usage**

* Returns the sentence on which the parser is working

– **Returns** - An array of words on which the parser is working• *Parse*

```
public void Parse( )
```

– **Usage**

* Try to parse the sentence in **arguments**. The method also interprets the sentence and calls the appropriate method in the appropriate interface. When the sentence could not be parsed, an exception is thrown.

– **Exceptions**

* **SyntaxInvalidException** - Thrown when the sentence could not be parsed.

3.25 Package aCommunicator

Package Contents *Page*

Classes	
ACommunicator	264
<i>This component has the function to communicate with the dispatcher.</i>	

9020

3.25.1 Classes

CLASS **ACommunicator**

This component has the function to communicate with the dispatcher.

9025 DECLARATION

```
public class ACommunicator
extends java.lang.Object
implements interfaces.ICommunicator
```

CONSTRUCTORS

9030

- *ACommunicator*
 public ACommunicator()
 - **Usage**
 - * Creates a new instance of this class

9035 METHODS

- *begin*
 public void begin()
 - **Usage**
 - * Method that is called when an instance has been created and the interfaces have been set
-
- *send*
 public GRP send(genericpackages.GSP pack, java.net.InetAddress dispatchAddr)
-
- *setInterfaces*
 public void setInterfaces()
 - **Usage**
 - * Sets references to all interfaces this component might need

3.26 Package dLogger

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
DLogger	266
<i>The function of this component is to log the actions of the applications.</i>	
TimeStamper	267
<i>Class that returns timestamps of the current time.</i>	
<hr/>	

9055

3.26.1 Classes

CLASS **DLogger**

9060 The function of this component is to log the actions of the applications. It is capable of logging regular messages and error messages separately.

DECLARATION

```
public class DLogger
extends java.lang.Object
```

9065

CONSTRUCTORS

- *DLogger*
 public DLogger()

METHODS

- *debug*
 public static void debug(java.lang.String msg)
 - **Usage**
 * Prints a message to standard output. Should be used only when the message is debug related.
 - **Parameters**
 * *msg* - The message to be printed

9080

-
- *error*
 public static void error(java.lang.String msg)
 - **Usage**
 * Writes a string to the error log file, prefixed with the current date and time
 - **Parameters**
 * *msg* - Message to write

9085

-
- *error*
 public static void error(java.lang.String msg, boolean stop)
 - **Usage**

9090 * Writes a string to the error log file, prefixed with the current date and time.
 Additionally it checks the boolean whether or not to abort execution of the
 program.

– **Parameters**

9095 * **msg** - Message to write
 * **stop** - Whether or not the application should be terminated

• *exception*

`public static void exception(java.lang.Exception ex)`

– **Usage**

9100 * This method handles an exception, by printing the stacktrace to the screen.

– **Parameters**

 * **ex** - The exception to be handled.

• *exception*

9105 `public static void exception(java.lang.Exception ex, boolean stop)`

– **Usage**

 * This method handles an exception, by printing the stacktrace to the screen.
 Additionally, it stops the execution of the application if the boolean is set to
 true.

9110 – **Parameters**

 * **ex** - The exception to be handled.

• *log*

`public static void log(java.lang.String msg)`

9115 – **Usage**

 * Writes a string to the log file, prefixed with the current date and time

– **Parameters**

 * **msg** - Message to write

CLASS **TimeStamper**

9120

Class that returns timestamps of the current time.

DECLARATION

<pre>public class TimeStamper extends java.lang.Object</pre>

9125

CONSTRUCTORS

- *TimeStamper*
`public TimeStamper()`

9130 METHODS

- *getDateStamp*
`public static String getDateStamp()`
 - **Usage**
 - * Returns the current date in the form yyyy/MM/dd
 - **Returns** - The current date as **String** in format yyyy/MM/dd

-
- *getDateTimeStamp*
`public static String getDateTimeStamp()`
 - **Usage**
 - * Returns the current date in the form yyyy/MM/dd HH:mm:ss
 - **Returns** - The current date and time as **String** in format yyyy/MM/dd HH:mm:ss

-
- *getTimeStamp*
`public static String getTimeStamp()`
 - **Usage**
 - * Returns the current time in the form HH:mm:ss
 - **Returns** - The current time as **String** in format HH:mm:ss

9150

3.27 Package dCommunicator

<i>Package Contents</i>	<i>Page</i>
<hr/>	
Classes	
DCommunicator	270
<i>This component has the function to communicate with the agent.</i>	
<hr/>	

3.27.1 Classes

9155 CLASS **DCommunicator**

This component has the function to communicate with the agent. This is one of the two components in the dispatcher application able to communicate with the outside world.

DECLARATION

9160

```
public class DCommunicator
extends java.lang.Object
implements java.lang.Runnable
```

CONSTRUCTORS

- 9165
 - *DCommunicator*
`public DCommunicator()`

METHODS

- 9170
 - *await*
`public void await()`
 - **Usage**
 - * This method awaits a request over the network. When it gets a request, it starts a thread called `ProcessConnection`, then waits again.

9175
 - *begin*
`public void begin()`
 - **Usage**
 - * Method that is called when an instance has been created and the interfaces have been set

9180
 - *decrConnections*
`public void decrConnections()`
 - **Usage**
 - * There's a maximum connection count, this method should be called if a connection is closed.

9185
 - *incrConnections*
`public boolean incrConnections()`

– **Usage**

9190 * There's a maximum connection count, this method should be called if a
 connection is opened.

• *run*

public void run()

9195 – **Usage**

 * Starts await.

• *setInterfaces*

public void setInterfaces(interfaces.IDPackageInput idPackageInput)

9200 – **Usage**

 * Sets references to all interfaces this component might need

– **Parameters**

 * idPackageInput -

3.28 Package <none>

Package Contents Page

9205

Classes

Agent273

...no description...

Client273

...no description...

Dispatcher.....274

...no description...

9210

3.28.1 Classes

CLASS **Agent**

9215 DECLARATION

```
public class Agent
extends java.lang.Object
```

CONSTRUCTORS

9220

- *Agent*
 public Agent()

METHODS

9225

- *main*
 public static void main(java.lang.String [] args)

CLASS **Client**

DECLARATION

9230

```
public class Client
extends java.lang.Object
```

CONSTRUCTORS

9235

- *Client*
 public Client()

METHODS

9240

- *main*
 public static void main(java.lang.String [] args)

CLASS **Dispatcher**

DECLARATION

9245 `public class Dispatcher`
`extends java.lang.Object`

CONSTRUCTORS

9250

- *Dispatcher*
`public Dispatcher()`

METHODS

- *main*
`public static void main(java.lang.String [] args)`

Chapter 4

Build procedure

The build and install procedure are documented in [SUM, section 3.1].

Appendix A

Source code listings

9260 A.1 Client Components

Component	Location(s)
commands	Client/src/commands/ClientCommands.java
exceptions	Client/src/exceptions/CommandSyntaxInvalidException.java Client/src/exceptions/CommandArgumentException.java Client/src/exceptions/InvalidIdentifierException.java Client/src/exceptions/SPINGRIDException.java
cLogger	Client/src/cLogger/TimeStamper.java Client/src/cLogger/CLogger.java
cCommunicator	Client/src/cCommunicator/CCommunicator.java
interfaces	Client/src/interfaces/ICDataProviderCommands.java Client/src/interfaces/ICJobProviderCommands.java Client/src/interfaces/ICInputProcessor.java Client/src/interfaces/ICSystemAdminCommands.java Client/src/interfaces/ICCommunicator.java Client/src/interfaces/ICApplicationProviderCommands.java Client/src/interfaces/ICOutputProcessor.java Client/src/interfaces/ICUserCommands.java Client/src/interfaces/ICProjectAdminCommands.java
HTTPMessage	Client/src/HTTPMessage/HTTPResponse.java Client/src/HTTPMessage/HTTPRequest.java Client/src/HTTPMessage/KeyValues.java
configuration	Client/src/configuration/Options.java
cIOProcessor	Client/src/cIOProcessor/CommandArgument.java Client/src/cIOProcessor/CIOProcessor.java Client/src/cIOProcessor/CommandErrors.java Client/src/cIOProcessor/CommandParser.java
genericpackages	Client/src/genericpackages/GPType.java Client/src/genericpackages/GRP.java Client/src/genericpackages/GSP.java

cTranslator	Client/src/cTranslator/CTranslator.java
Client	Client/src/Client.java

A.2 Agent Components

Component	Location(s)
aLogger	Agent/src/aLogger/TimeStamper.java Agent/src/aLogger/ALogger.java
exceptions	Agent/src/exceptions/ApplicationNotFoundException.java Agent/src/exceptions/ArgumentException.java Agent/src/exceptions/IntervalException.java Agent/src/exceptions/KeywordException.java Agent/src/exceptions/CommunicationException.java Agent/src/exceptions/SyntaxInvalidException.java Agent/src/exceptions/SPINGRIDException.java
interfaces	Agent/src/interfaces/IAJobSchedulerForDataManager.java Agent/src/interfaces/IADistributorForIOProcessor.java Agent/src/interfaces/IAJobExecutor.java Agent/src/interfaces/IAJobSchedulerForJobExecutor.java Agent/src/interfaces/IADistributorForJobScheduler.java Agent/src/interfaces/IAJobSchedulerForDistributor.java Agent/src/interfaces/IAInputProcessor.java Agent/src/interfaces/IACommunicator.java Agent/src/interfaces/IAOutputProcessor.java Agent/src/interfaces/IADDataManager.java Agent/src/interfaces/IAResourceProviderCommands.java
HTTPMessage	Agent/src/HTTPMessage/HTTPRequest.java Agent/src/HTTPMessage/KeyValues.java Agent/src/HTTPMessage/HTTPResponse.java
aTranslator	Agent/src/aTranslator/ATranslator.java
aJobExecutor	Agent/src/aJobExecutor/StreamGobbler.java Agent/src/aJobExecutor/cmdExec.java Agent/src/aJobExecutor/AJobExecutor.java
JSDL	Agent/src/JSDL/JSDLComplexRange.java Agent/src/JSDL/JSDLApplication.java Agent/src/JSDL/JSDLSpingridOperatingSystem.java Agent/src/JSDL/JSDLSpingridApplication.java Agent/src/JSDL/JSDLFileSystem.java Agent/src/JSDL/JSDLJobIdentification.java Agent/src/JSDL/JSDLDataStaging.java Agent/src/JSDL/JSDLSimpleRange.java Agent/src/JSDL/JSDLResources.java Agent/src/JSDL/JSDLOperatingSystem.java Agent/src/JSDL/JSDL.java Agent/src/JSDL/JSDLSpingridExecutable.java

	Agent/src/JSDL/JSDLDataStagingList.java Agent/src/JSDL/JSDLLowerBoundedRange.java Agent/src/JSDL/JSDLSpingridDatasets.java
commands	Agent/src/commands/AgentCommands.java
aCommunicator	Agent/src/aCommunicator/ACommunicator.java
types	Agent/src/types/PollResponse.java Agent/src/types/Interval.java Agent/src/types/PollRequest.java Agent/src/types/ResourceProperties.java Agent/src/types/AgentSpecs.java Agent/src/types/DownOrUploadStatus.java
aDataManager	Agent/src/aDataManager/ADDataManager.java Agent/src/aDataManager/ReadLine.java Agent/src/aDataManager/HTTPUpload.java Agent/src/aDataManager/Download.java Agent/src/aDataManager/FTPUpload.java Agent/src/aDataManager/Upload.java
aIOProcessor	Agent/src/aIOProcessor/CommandArgument.java Agent/src/aIOProcessor/AIOProcessor.java Agent/src/aIOProcessor/CommandParser.java Agent/src/aIOProcessor/CommandErrors.java
aJobScheduler	Agent/src/aJobScheduler/AJobScheduler.java
genericpackages	Agent/src/genericpackages/GPType.java Agent/src/genericpackages/GRP.java Agent/src/genericpackages/GSP.java
aDistributor	Agent/src/aDistributor/ADistributor.java
configuration	Agent/src/configuration/Options.java
Agent	Agent/src/Agent.java

A.3 Dispatcher Components

Component	Location(s)
commands	Dispatcher/src/commands/AgentCommands.java Dispatcher/src/commands/ClientCommands.java
exceptions	Dispatcher/src/exceptions/RoleException.java Dispatcher/src/exceptions/TrustException.java Dispatcher/src/exceptions/SyntaxInvalidException.java Dispatcher/src/exceptions/KeywordException.java Dispatcher/src/exceptions/DataSetException.java Dispatcher/src/exceptions/ProjectException.java Dispatcher/src/exceptions/InvalidIdentifierException.java Dispatcher/src/exceptions/UserException.java Dispatcher/src/exceptions/AuthenticationException.java Dispatcher/src/exceptions/SPINGRIDException.java Dispatcher/src/exceptions/ApplicationException.java

	Dispatcher/src/exceptions/JobException.java
dDatabaseManager	Dispatcher/src/dDatabaseManager/DDatabaseManager.java
dLogger	Dispatcher/src/dLogger/TimeStamper.java Dispatcher/src/dLogger/DLogger.java
interfaces	Dispatcher/src/interfaces/IDInputProcessor.java Dispatcher/src/interfaces/IDQueryHandler.java Dispatcher/src/interfaces/IDPackageInput.java Dispatcher/src/interfaces/IDResourceProviderCommands.java Dispatcher/src/interfaces/IDDataProviderCommands.java Dispatcher/src/interfaces/IDJobProviderCommands.java Dispatcher/src/interfaces/IDACommandInput.java Dispatcher/src/interfaces/IDSystemAdminCommands.java Dispatcher/src/interfaces/IDOutputProcessor.java Dispatcher/src/interfaces/IDApplicationProviderCommands.java Dispatcher/src/interfaces/IDUserCommands.java Dispatcher/src/interfaces/IDCCommandInput.java Dispatcher/src/interfaces/IDProjectAdminCommands.java
HTTPMessage	Dispatcher/src/HTTPMessage/HTTPResponse.java Dispatcher/src/HTTPMessage/HTTPRequest.java Dispatcher/src/HTTPMessage/KeyValues.java
dCommunicator	Dispatcher/src/dCommunicator/ProcessConnection.java Dispatcher/src/dCommunicator/DCommunicator.java
types	Dispatcher/src/types/PollResponse.java Dispatcher/src/types/PollRequest.java
dIOProcessor	Dispatcher/src/dIOProcessor/CommandParseErrors.java Dispatcher/src/dIOProcessor/CommandParser.java Dispatcher/src/dIOProcessor/DIOProcessor.java
genericpackages	Dispatcher/src/genericpackages/GPType.java Dispatcher/src/genericpackages/GRP.java Dispatcher/src/genericpackages/GSP.java
dAgentManager	Dispatcher/src/dAgentManager/QueryCreator.java Dispatcher/src/dAgentManager/DAgentManager.java
dClientManager	Dispatcher/src/dClientManager/QueryCreator.java Dispatcher/src/dClientManager/EMessages.java Dispatcher/src/dClientManager/DClientManager.java
dTranslator	Dispatcher/src/dTranslator/DTranslator.java
configuration	Dispatcher/src/configuration/Options.java
Dispatcher	Dispatcher/src/Dispatcher.java
JSDL	Dispatcher/src/JSDL/JSDLSpingridApplication.java Dispatcher/src/JSDL/JSDLDataStagingList.java Dispatcher/src/JSDL/JSDLApplication.java Dispatcher/src/JSDL/JSDLJobIdentification.java Dispatcher/src/JSDL/JSDLComplexRange.java Dispatcher/src/JSDL/JSDLFileSystem.java Dispatcher/src/JSDL/JSDLOperatingSystem.java Dispatcher/src/JSDL/JSDLDataStaging.java

Dispatcher/src/JSDL/JSDLSimpleRange.java
Dispatcher/src/JSDL/JSDLLowerBoundedRange.java
Dispatcher/src/JSDL/JSDLResources.java
Dispatcher/src/JSDL/JSDLSpinggridDatasets.java
Dispatcher/src/JSDL/JSDL.java
Dispatcher/src/JSDL/JSDLSpinggridOperatingSystem.java
Dispatcher/src/JSDL/JSDLSpinggridExecutable.java

Appendix B

Requirements traceability matrix

9265 The requirements traceability matrix has not been made. This is because the document contains already too many pages. Furthermore, there was little time to make this matrix. The project manager knows about this and advised the SPINGRID team to make better use of the available time.