



IST-2001-32133

GridLab - A Grid Application Toolkit and Testbed

iGrid v1.3 Public Release Software Guide

Author(s):	Giovanni Aloisio, Massimo Cafaro, Italo Epicoco, Daniele Lezzi, Maria Mirto, Silvia Mocavero
Document Filename:	iGrid Guide
Work package:	GridLab WP10 - Information Services
Partner(s):	
Lead Partner:	University of Lecce, Italy
Config ID:	
Document classification:	

Abstract: This document describes the iGrid Information Service v1.3 Public Release. The documentation covers iGrid installation instructions, relational schema, information providers, the APIs (C language) to search for and to publish information





Contents

1	Introduction	2
2	Installation	3
2.1	iGrid installation	3
2.1.1	Installation of iGrid database on postgresSQL with SSL support	3
2.1.2	iGrid web service installation	4
2.2	iGrid update	7
2.3	Environment Variables	7
3	GridLab iGrid relational schema	9
3.1	Information provided by users and/or services	9
3.2	Information provided by the information providers (system info)	11
3.3	Information model	15
4	iGrid web service	27
4.1	Search	27
4.2	ServiceDescription	28
4.3	Services	28
4.4	web Services	30
4.5	Firewall	32
4.6	Virtual Organization	34
4.7	Local Resource Management System	35
4.8	CPU	36
4.9	Memory	36
4.10	Network	37
4.11	Operating System	37
4.12	Device	37
4.13	Certification Authority	38
4.14	User	38

1 Introduction

iGrid has been designed as a relational Information Service for discovering and monitoring static and dynamic information related to resources belonging to a computational grid, taking into account the requirements suggested by GridLab WPs. The iGrid architecture is based on iServes and iStores. An iServe gathers local information, while the iStore collects information related to all of the registered iServes. An iStore can register itself to other iStores, allowing the creation of complex, arbitrary hierarchies. When information needs to be propagated from an administrative domain to a different one, it is automatically compressed and decompressed as needed. Our GRelC Toolkit (Grid Relational Catalog) provides a set of libraries for data access and integration in a grid environment. iGrid relies heavily on GRelC (i) Standard Database Access Interface (libgrelecSDAI-v2.0) and (ii) MultiQuery (libgrelecMultiQuery-v1.0). The former provides a set of primitives to transparently get access to and interact with different data sources. It offers an uniform access interface to several DBMSs exploiting a set of wrappers (dynamic libraries providing dynamic binding to specific DBMSs) which can be easily plugged into our system. The SDAI library takes into account and handles DBMS heterogeneity (different back-end errors, APIs and data types), hiding all of the differences by means of a uniform layer. To date, SDAI wrappers are developed for PostgreSQL, MySQL and unixODBC data sources, providing interaction both with relational and not relational databases. This library provides core functionalities such as: (i) open/close connection with database, (ii) submit query to database, (iii) open/commit/abort/rollback transaction, (iv) lock/unlock table, (v) extract value from the retrieved recordset, (vi) etc. The MultiQuery library (built on top of libgrelecSDAI) is used to insert (single shot) a huge amount of data (generated by the Information Providers) in the iGrid Relational Information System. Using the MultiQuery it is possible to:

1. compute (single shot) many insert/update/delete queries (indeed, in the same MultiQuery XML file it is possible to specify information related to several records of multiple tables);
2. reduce the interactions between Information Providers and the iGrid Server (indeed, for a MultiQuery submission there is only one feedback, whereas for N queries there are N feedbacks);
3. move to the iGrid Server side the entire computation (reducing the connection time between Information Providers and iGrid Server and speeding up the entire submission process).

The system is able to store two kind of information:

1. user and/or service supplied information;
2. system information (directly extracted from computational resources).

iGrid security is based on the Globus Toolkit GSI infrastructure; the iGrid web service has been secured using our GSI plug-in for gSOAP technology. The iGrid release includes:

- iGrid relational schema;
- Information providers for linux, Mac OS X, tru64;
- iGrid web service;
- iGrid sample clients.

The document is organized as follows: Section 2 provides the instructions to configure, build and install iGrid; Section 3 describes the information stored in the iGrid database and finally, Section 4 presents the web service methods related to information searching/publishing.

2 Installation

This package includes the iGrid relational schema and related system information providers; a web service and related sample clients are also included. To build and install the package please follow these installation instructions. If you are upgrading the igrid server from previous versions, you can skip to section 'iGrid Update'.

2.1 iGrid installation

Required software :

- Globus Toolkit v3.2.1 pre-ogsi (<http://www.globus.org>);
- gSOAP Toolkit v2.6.2 NOTE: iGrid requires THIS version, since newer releases have DIME support broken (<http://www.cs.fsu.edu/~engelen/soap.html>);
- postgresSQL v.7.4.x or newer (<http://www.postgresql.org>). Be aware that postgres functions library must be built with `-enable-thread-safety` configure option;
- libxml2 v2.6.7 or newer (<http://www.xmlsoft.org>);
- Zlib v1.2.1 or newer (<http://www.gzip.org/zlib/>);
- Apple CHUD tools v4.1.1 or newer Note: this software is required ONLY on Mac Os X; (ftp://ftp.apple.com/developer/Tool_Chest/Testing_-_Debugging/Performance_tools/)

To install on your platform please download gSOAP, and copy the files according to the following: gSOAP

- gSOAP compiler file (soapcpp2) -> bin
- gSOAP library file (stdsoap2.c) -> src/gsoap-gsi
- gSOAP header file (stdsoap2.h) -> include

2.1.1 Installation of iGrid database on postgresSQL with SSL support

1. Make sure that your postgres package has been built with SSL support and thread safe libraries (`-enable-thread-safety` and `-with-openssl` configure options)
2. Open the file postgresql.conf and set SSL on

```
ssl = true
```

3. Install a certificate for postmaster.
Postmaster demon needs a certificate. You can generate an X509v3 certificate and key using openssl (please refer to the openssl documentation).
4. Copy the certificate file into the postgres data directory and name it server.crt
5. Copy the key file into the postgres data directory and name it server.key

6. Copy the certificate file related to the certification authority that has signed the postgres certificate into the postgres data directory and name it root.crt. If you have generated a self signed certificate then root.crt and server.crt must be the same file
7. Create the "iGrid" database user (with permission to create database only) assigning a password to the new user

```
createuser -P <dbuser>
```

8. Create the "iGrid" database

```
createdb -O <dbuser> <dbname>
```

9. Append a new entry in the pg_hba.conf in the directory data located in the postgresSQL install dir

```
TYPE = hostssl  
DATABASE = <dbname>  
USER = <dbuser>  
METHOD = password
```

10. Restart the postgresSQL server
11. Associate the iGrid relational schema to the database
use the provided dump file (`${GRIDLAB_LOCATION}/schema/iRelGrid.sql`)

```
psql <dbname> -U <dbuser>  
-e <<iGrid.sql full path> 2>\&1 | tee iGrid-log
```

2.1.2 iGrid web service installation

1. Create (or modify) the `/etc/gridlab.conf` file according to your environment
The iGrid web service requires the following environment variables defined in the `gridlab.conf` file:

HOSTNAME the FQDN name of the host

GRIDLAB_LOCATION the iGrid installation directory

GLOBUS_FLAVOR_THREADS the name of the threaded flavor of Globus SDK

GLOBUS_FLAVOR the name of the non threaded flavor of Globus SDK

GLOBUS_LOCATION the Globus Toolkit installation directory

GLOBUS_INCLUDE_THREADS the directory containing header files for the threaded flavor of the Globus Toolkit

GLOBUS_INCLUDE the directory containing header files for the non threaded flavor of the Globus Toolkit

GRID_MAPFILE pathname of Globus grid-mapfile

CERTIFICATION_AUTHORITY_DIR pathname of directory containing recognized CAs certificates, usually `/etc/grid-security/certificates`

IGRID_CONF_FILENAME pathname of the iGrid configuration file; usually `${GRIDLAB_LOCATION}/etc/igrid.conf.xml`

GAS_URL the optional web service access URL to contact your GridLab Authorization service (GAS)

2. Configure the distribution

```
configure --help
```

This will provide you with configure options; please apply to the following the options you need. Relevant options include:

```
--with-listening-port
```

This option sets the web service listening port; defaults to 19000 (NOTE: this port is an officially IANA registered port for the iGrid service; see <http://www.iana.org/assignments/port-numbers>).

```
--with-authorized-dn
```

This option sets the pathname of the file `authorized_dn` used to authorize incoming connections through distinguished names; defaults to `etc/authorized_dn`. Please note that this pathname is not absolute, it is relative to the iGrid installation on `$GRIDLAB_LOCATION/igrid`.

```
--disable-ssl
```

Using this option you can disable the SSL binding to postgres database (by default SSL support is enabled).

```
--enable-mercury
```

This enables the logging service provided by Mercury Monitoring Service (by default it is disabled). For information about Mercury please refer to <http://www.gridlab.org/wp-11>

3. Configure

```
configure
```

This will configure the package using default options.

4. Build the software

```
make
```

5. Install everything as user with read/write permissions to the \$PREFIX directory

```
make install
```

6. Modify the iGrid configuration (etc/igrid.conf.xml) according to your environment
The following XML nodes can be modified:

Hostname FQDN of the machine where you are installing iGrid

SystemInfo_SD pathname of the system information spool directory; it can be absolute or relative to the iGrid server directory (e.g.: suppose you have installed iGrid on /opt/igrid, then the pathname can be relative to /opt/igrid/server)

UserInfo_SD pathname of the user/service information spool dir directory; it can be absolute or relative to the iGrid server directory

iNodeInfo_SD pathname of the iStore information spool directory; it can be absolute or relative to the iGrid server directory

RegistrationPeriod frequency (in seconds) to be used by iServe to send its information to registered iStores

PurgePeriod time frame (in seconds) during which an iStore the iServe is registered to, is temporarily removed from the local iStore registration list. This happens automatically when a registered iStore can not be contacted. After this time frame has expired, an iStore previously deleted is automatically inserted again in the registration list.

GrelC_Db_Hostname the host machine of iGrid database

GrelC_Db_Name name of iGrid database

GrelC_Db_Port TCP port of postgresQL DBMS as specified in the postgresql.conf in the directory data located in the postgresQL install dir

GrelC_Db_Login login of iGrid database user

GrelC_Db_Password password of the iGrid database user

The configuration file contains a list of local information providers characterized by:

Provider name name of the provider (each provider section must have a unique name).

Provider type type of the provider: 1 = executable; 2 = shared module (not yet supported)

Module pathname of the information provider executable.

Arguments specific arguments to be passed to the information provider.

InfoTTL validity time of information generated by the information provider.

Moreover, the configuration file contains a list of the hosts allowed to contact the local iGrid web service.

iNode_DN Distinguished Name of hosts/services allowed to store their information to the local iGrid web service. Note that if GAS authorization mechanism is selected, this field will be ignored.

iStore_WS access url of the iStore web service where local information must be sent

7. Run the server using a non privileged user account. In order to succeed, you will need to create in the home directory of the user that will be used for running the web service a .globus directory containing the host machine credential. Please, be sure to set appropriate permissions (.globus directory: 755; host certificate: 444; host key: 400).

```
cd $GRIDLAB_LOCATION/igrid/server ./igrid
```

```
USAGE: ./igrid
-p <num> set service port number
-f <file> configuration file
-d <level> debug level
-v version
-l <file> standard output redirection
-e <file> standard error redirection
```

To stop the server, send to the igrid process the SIGTERM signal.

2.2 iGrid update

If you are updating from a previous version you do not need to rebuild the entire database: the following command will update it

```
./iRelGrid_update2.scr <dbname> <dbuser> <dbport> <iRelGrid.sql_path>
```

<dbname> name of the iGrid database(specified in a.2)

<dbuser> login of the iGrid database user(specified in a.1)

<dbport> TCP port of postgresQL DBMS as specified in the postgresql.conf in the directory data located in the postgresQL install dir

<iRelGrid.sql_path> full path of the iRelGrid.sql file in the iGrid2/schema directory

you can now build the distribution and install it following the standard procedure:

-configure (for more information you can refer to the previous section)

-make

-make install

2.3 Environment Variables

In this section we list all of the environment variables that the iGrid server checks at runtime:

IGRID_CONF_FILENAME pathname of the iGrid configuration file; usually {prefix}/etc/igrid_conf.xml

IGRID_DEBUG it allows setting the debug level; 0 = No Debug, 1 = low, 2 = medium, 3 = high

IGRID_MAPFILE pathname of trusted users distinguished name list

GAS_URL access URL of the GAS Authorization service. For more information about GAS please refer to <http://www.gridlab.org/wp-6>

IGRID_AUTH = [gas | gas_only | local]: this variable must be used to choose the authorization mechanism to be used by iGrid web service.

gas_only The iGrid service will contact the GAS service to get the authorization decision for the user. If GAS service is not available or unreachable the user will not be authorized to access iGrid. In this case the field <Allowed> inside the iGrid configuration file and the local authorized_dn file are ignored;

gas (default) The iGrid service will contact the GAS service to get the authorization decision for the user. If the GAS service is not available or unreachable the local authorization mechanism will be used instead;

local the authorization mechanism is based on a local Access Control List.

Be aware that the authorization mechanism is valid both for user authorization and also to authorize iServe nodes to store their information on iStore nodes.

3 GridLab iGrid relational schema

This section describes the iGrid information schema. The schema closely takes into account the requirements for grid computing provided by GridLab WPs as described initially in the deliverable D10.2 and as required during the course of the project. Of course, this is not meant to be static, the schema will continue to evolve and will be extended to support additional information that will be required by the GridLab project or grid community. The current Public Release version 1.1 provides information related to:

- User/service supplied information.
 - Services;
 - Web Services;
 - Firewalls;
 - Virtual Organizations;
- System information.
 - Network interfaces;
 - Operating System;
 - Memory;
 - Cpu;
 - Local Resource Management System;
 - Gatekeeper Service;
 - Devices;
 - Trusted Certification Authorities;
 - Users.

System information is directly provided by related information providers. User/service information is provided by user and/or services through the iGrid web service. This can be done calling the `gridlab-*-register` methods.

We now briefly review the attributes of the iGrid schema.

3.1 Information provided by users and/or services

During the course of the GridLab project and in the grid community, a number of services and web services will be developed. One of the most important requirements for grid computing scenarios is the ability to discover services and web services dynamically. The iGrid system provides developers with the following functionalities: register, unregister, update and lookup. More than one instance for each service or web service can be registered. The following information belongs to these categories:

- Service information.
 - GridLab-iGrid-Service-id: service identification number;
 - GridLab-iGrid-Service-name: service logical name;

- GridLab-iGrid-Service-description: service description;
- GridLab-iGrid-Service-keywords: set of keywords of a service;
- GridLab-iGrid-Service-defaultport: service default port.
- Service instance information.
 - GridLab-iGrid-Service-accessurl: service access URL;
 - GridLab-iGrid-Service-publisher: service publisher (X509v3 certificate distinguished name);
 - GridLab-iGrid-Service-creationdate: creation date of the service instance information;
 - GridLab-iGrid-Service-validitytime: validity time of the service instance information.
- Web Service information.
 - GridLab-iGrid-WebService-id: web service identification number;
 - GridLab-iGrid-WebService-name: web service logical name;
 - GridLab-iGrid-WebService-description: web service description;
 - GridLab-iGrid-WebService-keywords: set of keywords of a web service;
 - GridLab-iGrid-WebService-wsdlloc: (multivalue) URL where the WSDL document for the web Service can be found.
- web Service instance information.
 - GridLab-iGrid-WebService-accessurl: web service access URL;
 - GridLab-iGrid-WebService-publisher: web service publisher (X509v3 certificate distinguished name);
 - GridLab-iGrid-WebService-creationdate: creation date of the web service instance information;
 - GridLab-iGrid-WebService-validitytime: validity time of the web service instance information.

Information related to firewalls is strictly related to service information. As a matter of fact, before registering a service, developers will query iGrid to know dynamically the range of open ports available on a specified computational resource. This is required to allow other people to connect to a service. The following information belongs to this category:

- Firewall information.
 - GridLab-iGrid-Firewall-id: firewall identification number;
 - GridLab-iGrid-Firewall-hostname: firewall hostname;
 - GridLab-iGrid-Firewall-ports: (multivalue) attribute composed by the open ports (GridLab-iGrid-Firewall-port), the time frame during which each port (or a range of ports) is open (GridLab-iGrid-Firewall-firvaliditytime), the protocol (TCP/UDP) used to connect to firewall ports (GridLab-iGrid-Firewall-prot), the time indicating when each port (or a range of ports) begins to be open (GridLab-iGrid-Firewall-fircreationdate) and a flag (GridLab-iGrid-Firewall-type_por) to specify if binding to the port(or range of ports) is allowed;

- GridLab-iGrid-Firewall-adminDN: distinguished name of the firewall administrator;
- GridLab-iGrid-Firewall-publisher: firewall publisher (X509v3 certificate distinguished name);
- GridLab-iGrid-Firewall-creationdate: creation date of the firewall information;
- GridLab-iGrid-Firewall-validityTime: validity time of the firewall information.

The GridLab project spans multiple Virtual Organizations. Corresponding information will allow people to know, for instance, how to request an account on a machine belonging to a particular Virtual Organization, or the people to contact in case of trouble. The following information belongs to this category:

- Virtual Organization information.
 - GridLab-iGrid-Vo-id: Virtual Organization identification number;
 - GridLab-iGrid-Vo-name: Virtual Organization to which a specified computational resource belongs to;
 - GridLab-iGrid-Vo-resourceType: Virtual Organization resource type;
 - GridLab-iGrid-Vo-helpDeskPhoneNumber: (multivalue) help desk phone number;
 - GridLab-iGrid-Vo-helpDeskURL: URL pointing to a Virtual Organization web page;
 - GridLab-iGrid-Vo-adminName: administrator name of the VO;
 - GridLab-iGrid-Vo-jobmanager: jobmanager used by the Virtual Organization;
 - GridLab-iGrid-Vo-queue: job queue of the Virtual Organization;
 - GridLab-iGrid-Vo-fileSystem: (multivalue) available file systems composed by the type (GridLab-iGrid-Vo-type) and the pathname (GridLab-iGrid-Vo-pathname);
 - GridLab-iGrid-Vo-publisher: publisher (X509v3 certificate distinguished name) of the Virtual Organization information;
 - GridLab-iGrid-Vo-creationdate: the time the machine starts to belong to the Virtual Organization;
 - GridLab-iGrid-Vo-validityTime: the time frame during which the machine belongs to the Virtual Organization.

3.2 Information provided by the information providers (system info)

System information allows complex brokering strategies: for instance, once the set of computing resources available to a user and their features are known to a broker, it is possible to choose carefully where to submit a job. The broker decision will be based on the information gathered from the iGrid system and on the job requirements. Moreover, accessing local resource management information is crucial for resource management etc. The current release provides support for the PBS local resource management system. The following information belongs to the system info category:

- Host information.
 - GridLab-iGrid-Host-id: Host identification number;
 - GridLab-iGrid-Host-hostname: the FQDN of the Host;
 - GridLab-iGrid-Host-domainname: the domain name of the Host;

- Network information.
 - GridLab-iGrid-Net-name: Network name;
 - GridLab-iGrid-Net-address: Network address;
 - GridLab-iGrid-Net-mask: Network mask;
 - GridLab-iGrid-Net-creationdate: creation date of the Network information;
 - GridLab-iGrid-Net-validityTime: validity time of the Network information.
- Operative system information.
 - GridLab-iGrid-System-id: Operating System identification number;
 - GridLab-iGrid-System-name: Operating System name;
 - GridLab-iGrid-System-version: Operating System version;
 - GridLab-iGrid-System-release: Operating System release;
 - GridLab-iGrid-System-machine: host platform;
 - GridLab-iGrid-System-creationdate: creation date of the Operative System information;
 - GridLab-iGrid-System-validityTime: validity time of the Operative System information.
- Memory dynamic information.
 - GridLab-iGrid-Memdyn-freeram: free ram space;
 - GridLab-iGrid-Memdyn-freeswap: free swap space;
 - GridLab-iGrid-Memdyn-creationdate: creation date of the dynamic Memory information;
 - GridLab-iGrid-Memdyn-validityTime: validity time of the dynamic Memory information.
- Memory static information.
 - GridLab-iGrid-Memsta-totalram: total ram space;
 - GridLab-iGrid-Memsta-totalswap: total swap space;
 - GridLab-iGrid-Memsta-creationdate: creation date of the static Memory information;
 - GridLab-iGrid-Memsta-validityTime: validity time of the static Memory information.
- Cpu dynamic information.
 - GridLab-iGrid-Cpudyn-number: Cpu internal identification number;
 - GridLab-iGrid-Cpudyn-load: Cpu load;
 - GridLab-iGrid-Cpudyn-uptime: Cpu user time;
 - GridLab-iGrid-Cpudyn-nicetime: Cpu nice time;
 - GridLab-iGrid-Cpudyn-systemtime: Cpu system time;
 - GridLab-iGrid-Cpudyn-idletime: Cpu idle time;
 - GridLab-iGrid-Cpudyn-creationdate: creation date of the dynamic Cpu information;
 - GridLab-iGrid-Cpudyn-validityTime: validity time of the dynamic Cpu information.

- Cpu static information.
 - GridLab-iGrid-Cpusta-id: Cpu identification number;
 - GridLab-iGrid-Cpusta-vendor: Cpu vendor;
 - GridLab-iGrid-Cpusta-version: Cpu version;
 - GridLab-iGrid-Cpusta-model: Cpu model;
 - GridLab-iGrid-Cpusta-cpumhz: Cpu Mhz;
 - GridLab-iGrid-Cpusta-cachesize: cache size;
 - GridLab-iGrid-Cpusta-features: Cpu features;
 - GridLab-iGrid-Cpusta-number: number of Cpu belonging to the host;
 - GridLab-iGrid-Cpusta-creationdate: creation date of the static Cpu information;
 - GridLab-iGrid-Cpusta-validityTime: validity time of the static Cpu information.
- Local Resource Management System information.
 - GridLab-iGrid-Lrms-id: Local Resource Management System identification number;
 - GridLab-iGrid-Lrms-type: the type of Local Resource Management System;
 - GridLab-iGrid-Lrms-version: the version of Local Resource Management System;
 - GridLab-iGrid-Lrms-creationdate: creation date of the Lrms information;
 - GridLab-iGrid-Lrms-validityTime: validity time of the Lrms information.

Each Local Resource Management System can manage one or more queues. Related information includes:

- Queue information.
 - GridLab-iGrid-Queue-id: the identification number of the Queue;
 - GridLab-iGrid-Queue-name: the name of the Queue;
 - GridLab-iGrid-Queue-assignedcpunumber: the number of cpus assigned to the Queue;
 - GridLab-iGrid-Queue-status: the Queue status;
 - GridLab-iGrid-Queue-maxqueueable: the max number of jobs allowed to reside in the Queue;
 - GridLab-iGrid-Queue-maxrunning: the max number of jobs allowed to run from this Queue;
 - GridLab-iGrid-Queue-queued: the number of jobs waiting in the Queue;
 - GridLab-iGrid-Queue-running: the number of running jobs in the cluster belonging to this Queue;
 - GridLab-iGrid-Queue-maxwallclocktime: the max wall clock time allowed for jobs submitted to the Queue in minutess;
 - GridLab-iGrid-Queue-maxcputime: the max CPU time allowed for jobs submitted to the Queue in minutes.

One or more job can be assigned to each queue. Related information includes:

- Job information.

- GridLab-iGrid-Job-id: the identification number of the Job;
 - GridLab-iGrid-Job-name: the name of the Job;
 - GridLab-iGrid-Job-jobid: the Job identifier string assigned by the primary scheduler;
 - GridLab-iGrid-Job-jobid2: the Job identifier string assigned by the secondary scheduler;
 - GridLab-iGrid-Job-owner: the SubjectName of the Job owner;
 - GridLab-iGrid-Job-reqcput: the cputime request of the Job in minutes;
 - GridLab-iGrid-Job-status: the status of the Job;
 - GridLab-iGrid-Job-submissiontime: the submission time of the Job;
 - GridLab-iGrid-Job-usedcputime: the consumed cputime of the Job in minutes;
 - GridLab-iGrid-Job-usedmem: the memory usage of the Job (in KB);
 - GridLab-iGrid-Job-usedwalltime: the consumed walltime of the Job in minutes;
 - GridLab-iGrid-Job-estimatedresponsetime: estimated time between Job submission and execution in seconds;
 - GridLab-iGrid-Job-description: Job description.
- Device dynamic information.
 - GridLab-iGrid-Devdyn-name: Device name;
 - GridLab-iGrid-Devdyn-freespace: Device free space;
 - GridLab-iGrid-Devdyn-creationdate: creation date of the dynamic Device information;
 - GridLab-iGrid-Devdyn-validityTime: validity time of the dynamic Device information.
- Device static information.
 - GridLab-iGrid-Devsta-name: Device name;
 - GridLab-iGrid-Devsta-mountpoint: Device mount point;
 - GridLab-iGrid-Devsta-size: Device size;
 - GridLab-iGrid-Devsta-fstype: filesystem type of the Device;
 - GridLab-iGrid-Devsta-creationdate: creation date of the static Device information;
 - GridLab-iGrid-Devsta-validityTime: validity time of the static Device information.
- Trusted Certification Authority information.
 - GridLab-iGrid-CA-id: Certification Authority identification number;
 - GridLab-iGrid-CA-version: Certification Authority version;
 - GridLab-iGrid-CA-serialnum: Certification Authority serial number;
 - GridLab-iGrid-CA-sigalg: Certification Authority signature algorithm;
 - GridLab-iGrid-CA-issuer: Certification Authority issuer;
 - GridLab-iGrid-CA-validfrom: Beginning date of the certificate validity (timestamp UTC);
 - GridLab-iGrid-CA-validto: End date of the certificate validity (timestamp UTC);

- GridLab-iGrid-CA-pkalg: Public key algorithm of the certificate;
 - GridLab-iGrid-CA-rsapk: RSA public key of the certification authority;
 - GridLab-iGrid-CA-subj: subject of the certification authority;
 - GridLab-iGrid-CA-certificate: the certificate itself;
 - GridLab-iGrid-CA-crlurl: Crl URL;
 - GridLab-iGrid-CA-modulus: Public key of the certificate.
- Information about User.
 - GridLab-iGrid-User-id: user’s identification number;
 - GridLab-iGrid-User-login: user’s login;
 - Distinguished Name information.
 - GridLab-iGrid-Dn-id: Distinguished Name identification number;
 - GridLab-iGrid-Dn-dn: Distinguished Name subject;

Each Host-User pair is characterized by the following information:

- Host-User information.
 - GridLab-iGrid-DnUsHo-path: user’s home directory path on the host;
 - GridLab-iGrid-DnUsHo-shell: user’s shell on the host;
 - GridLab-iGrid-DnUsHo-uid: user’s UID on the host;
 - GridLab-iGrid-DnUsHo-creationdate: creation date of the tern Host-DN-User information;
 - GridLab-iGrid-DnUsHo-validityTime: validity time of the tern Host-DN-User information.

3.3 Information model

We now present the iGrid data dictionary.

Table name	Definition	Obligation	Maximum size	Data type
TB_HOST	Contains the information about host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_host	Host machine table id (primary key)	Mandatory	4	Serial
host_name	FQDN of the host	Mandatory	256	Character
domain_name	Domain name of the host	Mandatory	256	Character
id_hos_sys	Primary key of the TB_SYSTEM table (foreign key)	Optional	4	Integer

id_hos_fir	Primary key of the TB_FIREWALL table (foreign key)	Optional	4	Integer
TB_NET	Contains the information about the network	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_net_inf	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
name	Interface name (primary key)	Mandatory	64	Character
address	Network interface address	Mandatory	15	Character
mask	Network interface mask	Optional	15	Character
creation_date	Creation date (timestamp UTC) of net information	Mandatory	8	bigint
validity_time	Validity time (sec) of net information	Mandatory	4	numeric
TB_CPU_STATIC	Contains static information about cpu	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_cpu_s	Cpu static table id (primary key)	Mandatory	4	Serial
number	Processor number	Mandatory	4	Integer
vendor	Processor vendor	Mandatory	128	Character
version	Processor version	Mandatory	32	Character
model	Processor model	Mandatory	64	Character
cpu_mhz	Processor clock speed (MHz)	Mandatory	4	Real
cache_size	Cache size (KB)	Mandatory	4	Integer
features	Processor feature and option flags	Optional	256	Character
creation_date	Creation date (timestamp UTC) of net information	Mandatory	8	bigint
validity_time	Validity time (sec) of cpu static information	Mandatory	4	numeric
id_cpu_s_hos	HOST machine table id (foreign key)	Mandatory	4	Integer

TB_SYSTEM	Contains the information about the operating system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_sys	System table id (primary key)	Mandatory	4	Serial
name	Operating system name	Mandatory	50	Character
release	Operating system release	Mandatory	50	Character
version	Operating system version	Mandatory	256	Character
machine	Operating system platform type	Mandatory	30	Character
creation_date	Creation date (timestamp UTC) of system information	Mandatory	8	bigint
validity_time	Validity time (sec) of system information	Mandatory	4	numeric
TB_DEVICE_STATIC	Contains the information about the static file system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_dev_s	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
name	Device static name (primary key)	Mandatory	256	Character
mount_point	File system mount point	Mandatory	256	Character
fs_type	Type of system	Mandatory	32	Character
size	Total space of the file system (MB)	Mandatory	8	bigint
creation_date	Creation date (timestamp UTC) of device static information	Mandatory	8	bigint
validity_time	Validity time (sec) of device static information	Mandatory	4	numeric
TB_DEVICE_DYNAMIC	Contains the information about the dynamic file system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object

id_hos_dev_d	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
name	Device dynamic name (primary key)	Mandatory	256	Character
free_space	File system free space (MB)	Mandatory	8	bigint
creation_date	Creation date (timestamp UTC) of device dynamic information	Mandatory	8	bigint
validity_time	Validity time (sec) of device dynamic information	Mandatory	4	numeric
TB_MEMORY_STATIC	Contains the information about the static memory	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_mem_s	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
totalRam	Total RAM (MB)	Mandatory	4	Real
totalSwap	Total Swap (MB)	Optional	4	Real
creation_date	Creation date (timestamp UTC) of memory static information	Mandatory	8	bigint
validity_time	Validity time (sec) of memory static information	Mandatory	4	numeric
TB_MEMORY_DYNAMIC	Contains the information about the dynamic memory	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_mem_d	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
freeRam	Free RAM (MB)	Mandatory	4	Real
freeSwap	Free Swap (MB)	Optional	4	Real
creation_date	Creation date (timestamp UTC) of memory dynamic information	Mandatory	8	bigint

validity_time	Validity time (sec) of memory dynamic information	Mandatory	4	numeric
TB_CPU_DYNAMIC	Contains dynamic information about cpu	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_hos_cpu_d	Host machine table id (primary key and foreign key)	Mandatory	4	Serial
number	Cpu number	Mandatory	4	Integer
load	Cpu load (%)	Mandatory	4	Real
user_time	User time (%)	Optional	4	Real
nice_time	Nice time (%)	Optional	4	Real
system_time	System time (%)	Optional	4	Real
idle_time	Idle time (%)	Optional	4	Real
creation_date	Creation date (timestamp UTC) of cpu dynamic information	Mandatory	8	bigint
validity_time	Validity time (sec) of cpu dynamic information	Mandatory	4	numeric
TB_FIREWALL	Contains information about the firewall hostname	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_firewall	Firewall table id (primary key)	Mandatory	4	Serial
hostname	Firewall hostname	Mandatory	256	Character
admin_dn	Administrator Distinguished Name	Optional	256	Character
publisher	Distinguished Name of the publisher	Mandatory	256	Character
creation_date	Creation date (timestamp UTC) of firewall information	Mandatory	8	bigint
validity_time	Validity time (sec) of firewall information	Mandatory	4	numeric
TB_FIR_PORTS	Contains information about ports range of the firewall hostname	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object

id_fir_por	Firewall table id (primary key and foreign key)	Mandatory	4	Integer
ports	Open ports range (primary key) in the following format: <dddddd>:<dddddd>	Mandatory	13	Character
protocol	Protocol UDP/TCP	Mandatory	16	Character
creation_date	Creation date (timestamp UTC) of firewall open ports range	Mandatory	8	bigint
validity_time	Validity time (sec) of firewall open ports range	Mandatory	4	numeric
type_por	a flag (GridLab-iGrid-Firewall-type_por) to specify if the binding to the port(or range of ports) has allowed	Mandatory	4	numeric
TB_VO	Contains information about the Virtual Organization	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_vo	Virtual Organization table id (primary key)	Mandatory	4	Serial
name	VO name	Mandatory	256	Character
admin_name	VO Administrator name	Optional	256	Character
resource_type	VO resource type	Optional	32	Character
job_manager	VO job manager	Optional	50	Character
queue	VO queue	Optional	50	Character
help_desk_url	URL pointing to a web page describing the VO	Optional	256	Character
RL_HOS_VO	Contains information about the relation between Virtual Organization and Host table	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object

id_rl_hos	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_vo	VO table id (primary key and foreign key)	Mandatory	4	Integer
publisher	Distinguished Name of the publisher	Mandatory	256	Character
creation_date	The time (timestamp UTC) the machine starts to belong to the Virtual Organization	Mandatory	8	bigint
validity_time	The time (sec) frame during which the machine belongs to the Virtual Organization	Mandatory	4	numeric
TB_VO_HDPN	Contains information about the help desk phone number of a Virtual Organization	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_vo_hdp	VO table id (primary key and foreign key)	Mandatory	4	Integer
help_desk_ph_num	Phone number related to the VO (primary key)	Mandatory	32	Character
TB_VO_FS	Contains information about the file system of a Virtual Organization	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_vo_fs	VO table id (primary key and foreign key)	Mandatory	4	Integer
path	Path of the VO file system (primary key)	Mandatory	256	Character
type	Type of the VO file system	Optional	256	Character
TB_LRMS	Contains information about the local resource management system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object

id_lrms	Local Resource Management System table id (primary key)	Mandatory	4	Serial
type	type of the LRMS	Mandatory	32	Character
version	version of the LRMS	Optional	16	Character
creation_date	Creation date (timestamp UTC) of LRMS information	Mandatory	8	bigint
validity_time	Validity time (sec) of LRMS information	Mandatory	4	numeric
id_hos_lrm	Host machine table id (foreign key)	Mandatory	4	Integer
TB_QUEUE	Contains information about the queues of a local resource management system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_queue	Queue table id (primary key)	Mandatory	4	Integer
name	Queue name	Mandatory	256	Character
asscpunum	Number of the cpus assigned to the queue	Optional	4	Integer
status	Queue status	Optional	32	Character
maxqueue	Max number of jobs allowed to reside in the queue	Optional	4	Integer
maxrun	Max number of jobs allowed to run from this queue	Optional	4	Integer
queued	Number of jobs waiting in the queue	Optional	4	Integer
running	Number of running jobs in the cluster belonging to this queue	Optional	4	Integer
maxwct	Max wall clock time allowed for jobs submitted to the queue in mins	Optional	4	Integer

maxcput	Max CPU time allowed for jobs submitted to the queue in mins	Optional	4	Integer
id_lrm_que	LRMS table id (foreign key)	Mandatory	4	Integer
TB_JOB	Contains information about the jobs of a local resource management system	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_job	Job table id (primary key)	Mandatory	4	Integer
name	Name of the job	Mandatory	256	Character
job_id	Job identifier assigned by lrms	Mandatory	64	Character
job_id2	Job identifier assigned by lrms	Optional	64	Character
owner	SubjectName of the job owner	Optional	256	Character
status	Status of the job	Optional	32	Character
reqcput	Cputime request of the job in minutes	Optional	4	Integer
subtime	Submission time (timestamp UTC) of the job	Optional	8	bigint
usedcput	Consumed cputime of the job in minutes	Optional	4	Integer
usedmem	Memory usage of the job	Optional	4	Integer
usedwt	Consumed wall-time of the job in minutes	Optional	4	Integer
estimatert	Estimated time between job submission till when job starts its execution	Optional	4	Integer
id_que_job	Queue table id (foreign key)	Mandatory	4	Integer
TB_CA	Contains information about the Certification Authority	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_ca	CA table id (primary key)	Mandatory	4	Integer

version	Version of the certificate	Optional	16	Character
serial_num	Serial Number of the certificate	Optional	16	Character
sig_alg	Signature algorithm of the certificate	Optional	64	Character
issuer	Issuer of the certificate	Mandatory	256	Character
valid_from	Beginning date of the certificate validity (timestamp UTC)	Mandatory	8	bigint
valid_to	End date of the certificate validity (timestamp UTC)	Mandatory	8	bigint
pk_alg	Public key algorithm of the certificate	Optional	64	Character
rsa_pk	RSA public key of the certification authority	Optional	64	VarChar
subj	Subject name of the CA certificate	Mandatory	256	Character
crL_url	Crl URL	Optional	256	Character
modulus	Public key of the certificate	Optional	4196	VarChar
certificate	the certificate itself	Mandatory	8192	VarChar
RL_HOS_CA	Contains information about the relation between Certification Authority and host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host table id (primary key)	Mandatory	4	Integer
id_rl_ca	CA table id (primary key)	Mandatory	4	Integer
creation_date	Creation date (timestamp UTC) of information related to host and CA tables	Mandatory	8	bigint
validity_time	Validity time (sec) of information related to host and CA tables	Mandatory	4	numeric

TB_SERVICE	Contains information about the service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_service	Service table id (primary key)	Mandatory	4	Serial
name	Service name	Mandatory	256	Character
description	Service description	Optional	1024	VarChar
keywords	Service keywords	Optional	256	Character
def_port	Service default port	Mandatory	4	Integer
RL_HOS_SERVICE	Contains information about the service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_service	Service table id (primary key and foreign key)	Mandatory	4	Integer
accessurl	Service access URL (primary key)	Mandatory	256	Character
publisher	Distinguished Name of the publisher	Mandatory	256	Character
creation_date	Creation date (timestamp UTC) of service information	Mandatory	8	bigint
validity_time	Validity time (sec) of service information	Mandatory	4	numeric
TB_WEB_SERVICE	Contains information about the web service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_web_service	web Service table id (primary key)	Mandatory	4	Integer
name	web Service name	Mandatory	256	Character
description	Service description	Optional	1024	VarChar
keywords	Service keywords	Optional	256	Character

TB_WS_WSDL_LOC	Contains information about the service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_ws_wsdl	web Service table id (primary key and foreign key)	Mandatory	4	Integer
wsdl_locationurl	WSDL location file URL (primary key)	Mandatory	256	Character
RL_HOS_WS	Contains information about the service running on the host machine	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host machine table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_ws	web Service table id (primary key and foreign key)	Mandatory	4	Integer
accessurl	web Service access URL (primary key)	Mandatory	256	Character
publisher	Distinguished Name of the publisher	Mandatory	256	Character
creation_date	Creation date (timestamp UTC) of web service information	Mandatory	8	bigint
validity_time	Validity time (sec) of web service information	Mandatory	4	numeric
TB_USER	Contains information about the user	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_user	User table id (primary key)	Mandatory	4	Serial
login	User login	Mandatory	32	Character
TB_DN	Contains information about the Distinguished Name mapped on the user	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_dn	DN table id (primary key)	Mandatory	4	Serial

dn	DN mapped on the user	Mandatory	256	Character
RL_HOS_DN_USE	Contains information about the relation among Distinguished Name, user and host tables	Use obligation from referencing object	Use maximum size from referencing object	Data type from referencing object
id_rl_hos	Host table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_user	User table id (primary key and foreign key)	Mandatory	4	Integer
id_rl_dn	DN table id (primary key and foreign key)	Mandatory	4	Integer
uid	User ID	Optional	4	Integer
shell	User shell	Optional	256	Character
path	User home dir	Optional	256	Character
creation_date	Creation date (timestamp UTC) of information related to host, user and Distinguished Name tables	Mandatory	8	bigint
validity_time	Validity time (sec) of information related to host, user and Distinguished Name tables	Mandatory	4	numeric

4 iGrid web service

The web service has been developed as a threaded server in order to improve the performances; it provides the user with several methods that can be invoked to search, register, unregister, update and lookup information.

The namespace used by the web service is urn:igrid. The web service methods are described using web Service Description Language in Appendix; here, we describe (using the C language) gSOAP return values for remote methods (classifying the parameters as IN, OUT, INOUT):

4.1 Search

```
-int igrid_search_sql(xsd_string filter, xsd_string *result)
```

This method can be used to query the iServe or iStore database using the SQL SELECT statement.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

filter: SQL SELECT statement (it must terminate with a semicolon).

OUT parameters:

result: contains the searched information in a XML document.

-int igrid_search(xsd_string host, xsd_string *result)

This method can be used to get all of the information stored in a iStore/iServe node. It is possible to filter the result by hostname, in this case only information about the specified host is returned.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

filter: SQL string to search.

OUT parameters:

result: string containing the searched information.

4.2 ServiceDescription

-int igrid_getServiceDescription(void *_ , xsd_string *result)

this method can be used to get the description and status of the service
output:

result: the string of description

4.3 Services

-int igrid_register_service(xsd_string name, xsd_string hostname, xsd_int port, xsd_string protocol, xsd_int dport, xsd_string description, xsd_string keywords, xsd_int validity_time, xsd_int *result)

This method allows registering a new service or a an instance of an existing service.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the service to be registered

hostname: the FQDN of the server that provides the service (must be a symbolic address)

port: port number where the service will be listening on

protocol: communication protocol to be used to contact the service

dport: default port number where the service will be listening on

description: human readable description of the service

keywords: service keywords

validity_time: validity time of the service instance information. If 0, the supplied information will never expire

OUT parameters:

result: set to a value < 0 if an error occurs during the registration, 0 on success for the service first instance registration; set to 1 for successful registration of additional instances of the service.

If the service being registered already exists, the registration will fail.

```
-int igrid_unregister_service(xsd_string name, xsd_string hostname, xsd_int port, xsd_int allinst, xsd_int *result)
```

This method removes a service, a single instance of a service or all of the instances of a service. Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the service to be removed
hostname: the FQDN of the server that provides the service
port: port number where the service will be listening on
allinst: a flag set to 1 to remove all of the instances of the specified service, set to 0 to remove a single instance.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

If the service does not exist the method returns 0 anyway.

```
-int igrid_update_service(xsd_string name, xsd_string keywords, xsd_string description, xsd_string url, xsd_int dport, xsd_int validity_time, xsd_int * result)
```

This method allows updating name, description, keywords, default port and validity time of a service. In particular, if the service name is specified then description, keywords and default port can be changed, while if the service access url is set then the service validity time can be updated.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the service to be updated
keywords: the keywords of the service to be updated
description: the description of the service to be updated
url: the access url of the service instance to be updated
dport: the default port where the service to be updated is listening on
validity_time: validity time of the service instance information.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

-int igrid_lookup_service (xsd_string name, xsd_string hostname, xsd_int dport, xsd_int sport, xsd_string prot, xsd_string *result)

This method returns all of the information about the services stored on the local iStore server grouped by hostname. If any input parameters have been specified, the method returns the information about all of the services and the related instances matching parameters values. It is possible to search the information about one or more services specifying the service name, hostname, default port parameters and grouping them by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the service to be searched.
hostname: the hostname where the service to be searched is running on.
dport: the default port of the service to be searched.
sport: the port number where the service to be searched is listening on.
prot: the protocol of the service to be searched.

OUT parameters:

result: string containing searched information stored in the local iStore database.

4.4 web Services

-int igrid_register_webservice (xsd_string name, xsd_string wsdllocation, xsd_string description, xsd_string url, xsd_string keywords, xsd_int validity_time, xsd_int * result)

This method allows registering a new web service. The wsdllocation parameter is a list, space-separated, of web service WSDL file location URL variables; each variable must not contain spaces.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the new web service to be registered.
wsdllocation: a list, space-separated, of web service WSDL file location URL variables.
description: human readable description of the web service.
url: the web service access URL (note: the URL must contain a symbolic address).
keywords: web service keywords.
validity_time: validity time of the web service instance information.

OUT parameters:

result: set to a value < 0 if an error occurs during the registration, 0 on success when registering the first instance of a web service; 1 on success when registering additional instances of the web service.

If the web service being registered already exists, the registration will fail.

-int igrid_unregister_webservice (xsd_string name, xsd_string url, xsd_int allinst, xsd_int * result)

This method removes an existing web service, a single instance of a web service or all of the instances of a web service.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the web service to be removed
url: the access URL of the web service instance to be removed
allinst: a flag set to 1 to remove all of the instances of the specified service, set to 0 to remove a single instance.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

If the web service does not exist the method returns 0 anyway.

-int igrid_update_webservice (xsd_string name, xsd_string keywords, xsd_string description, xsd_string url, xsd_int validity_time, xsd_int * result)

This method allows updating name, description and keywords of a web service or the access url of an instance of a web service.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the web service to be updated.
keywords: the keywords of the web service to be updated.
description: the description of the web service to be updated.
url: the access url of the web service instance to be updated.
validity_time: validity time of the web service instance information to be updated.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

-int igrid_lookup_webservice (xsd_string name, xsd_string hostname, xsd_string *result)

This method returns all of the information about the web services stored on the local iStore server grouped by hostname. If the logical name and/or the hostname have been specified, the method returns the information about specified web services running on specified host and related instances grouping them by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the web service to be searched.

hostname: the hostname where the web service to be searched is running on.

OUT parameters:

result: string containing searched information stored in the local iStore database.

-int `igrid_register_wsdlloc` (xsd_string name, xsd_string wsdlloc, xsd_int * result)

This method allows registering one or more new wsdl locations url associated to a web service. The wsdlloc parameter is a list, space-separated, of web service WSDL file location URL variables; each variable must not contain spaces.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the new web service to be registered.

wsdlloc: a list, space-separated, of web service WSDL file location URL variables.

result: set to a value < 0 if an error occurs, 0 on success.

-int `igrid_unregister_wsdlloc` (xsd_string name, xsd_string wsdlloc, xsd_int * result)

This method allows unregistering one or more new wsdl locations url associated to a web service. The wsdlloc parameter is a list, space-separated, of web service WSDL file location URL variables; each variable must not contain spaces.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the logical name of the new web service to be registered.

wsdlloc: a list, space-separated, of web service WSDL file location URL variables.

result: set to a value < 0 if an error occurs, 0 on success.

4.5 Firewall

-int `igrid_register_firewall`(xsd_strcmn info, xsd_arr_port_info pinfo, xsd_int * result)

This method allows registering information about a firewall installed on the grid resource.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

info: struct containing host name of the firewall, Distinguished Name of the firewall administrator, host name of the machine where the firewall is installed, validity time of the firewall information.

pinfo: struct containing space-separated list of open ports (or range of ports separated by - symbol), the protocol used to bind to these ports, the creation date and related validity time.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

`-int igrid_unregister_firewall(xsd_string hostname, xsd_int * result)`

This method removes information about a registered firewall.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

hostname: name of the firewall.

OUT parameters:

result: set to -1 if an error occurs, 0 on success.

If the firewall information does not exist the method sets result to 0 anyway.

`-int igrid_update_firewall(xsd_streem info, xsd_arr_port_info pinfo, xsd_int * result)`

This method allows updating admin distinguished name and validity_time of the firewall, protocol, creation_date, validity time of the open ports range. In particular if the firewall host name is specified, the admin distinguished name and validity_time can be updated, while specifying the open ports in the ports field, it is possible to change protocol, creation_date and validity time related to the open ports range.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

info: struct containing the name of the firewall to be updated, the distinguished name of the administrator of the firewall to be updated, validity time of the firewall information.

pinfo: struct containing a space-separated list of open ports (or range of ports separated by - symbol), the protocol used to bind to these ports, the creation date and related validity time.

OUT parameters:

result: set to a value < 0 if an error occurs, 0 on success.

`-int igrid_lookup_firewall(xsd_string name, xsd_string *result)`

This method returns all of the information about registered firewalls grouped by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the hostname of the firewall to be searched.

OUT parameters:

result: string containing searched information stored in the local iStore database.

4.6 Virtual Organization

```
-int igrid_register_vo(xsd_string name, xsd_string helpDeskPN, xsd_string restype, xsd_string jobm, xsd_string queue, xsd_string fs_path, xsd_string helpDeskURL, xsd_string adminname, xsd_string host, xsd_int validity_time, xsd_int * result)
```

This method allows the user to register a Virtual Organization "owning" the grid resource. If the user specifies any host field, this value is set default to FQDN of the server that provides the iGrid web service .

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: name of the Virtual Organization.
helpDeskPN: space separated Help-Desk Phone number of the Virtual Organization.
restype: resource type of the Virtual Organization.
jobm: jobmanager used by the Virtual Organization.
queue: job queue of the Virtual Organization.
fs_path: space separated available file systems (type:pathname) for Virtual Organization.
helpDeskURL: the URL of the help-Desk.
adminname: Name of the Virtual Organization administrator.
host: the name of the host to be registered to the Virtual Organization.
validity_time: validity time of the Virtual Organization information.

OUT parameters:

result: set to -1 if an error occurs, 0 on success.

```
-int igrid_unregister_vo(xsd_string name, xsd_string hostname, xsd_int allinst, xsd_int * result)
```

This method removes information about a Virtual Organization.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: name of the virtual organization whose information must be removed.
hostname: host name of the machine belonging to the Virtual Organization.
allinst: a flag set to 1 to remove specified Virtual Organization information and the relations between it and all of the registered hosts, set to 0 to remove a single relation.

OUT parameters:

result: set to -1 if an error occurs, 0 on success.

If the VO information does not exist the method sets result to 0 anyway.

```
-int igrid_update_vo(xsd_string name, xsd_string restype, xsd_string jobm, xsd_string queue, xsd_string helpDeskURL, xsd_string adminname, xsd_string host, xsd_string validity_time,
```

xsd_int * result)

This method allows updating resource type, job manager, queue, help desk url, admin distinguished name and validity time of a VO. In particular, resource type, job manager, queue, help desk url and admin distinguished name can be updated if VO name is specified. Moreover, to update the validity time of the VO the user should specify a valid time value; if she specifies both validity time and host then the information of a VO associated at a given host will be updated, while if only validity time is specified, the host value is set by default to the FQDN of the server that provides the iGrid web service.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

- name: name of the Virtual Organization to be updated.
- restype: resource type of the Virtual Organization to be updated.
- jobm: jobmanager used by the Virtual Organization to be updated.
- queue: job queue of the Virtual Organization to be updated.
- helpDeskURL: the URL of the help-Desk of the Virtual Organization to be updated.
- adminname: Name of the administrator of the Virtual Organization to be updated.
- host: the name of the host belonging to the Virtual Organization.
- validity_time: validity time of the Virtual Organization information to be updated.

OUT parameters:

- result: set to a value < 0 if an error occurs, 0 on success.

-int igrid_lookup_vo(xsd_string name, xsd_string *result)

This method returns all of the information about the registered virtual organizations grouped by hostname or virtual organization name. If a name has been specified, the method returns the information about specified virtual organization grouping them by hostname or virtual organization name.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

- name: the name of the Virtual Organization to be searched.

OUT parameters:

- result: string containing searched information stored in the local iStore database.

4.7 Local Resource Management System

-int igrid_lookup_lrms (xsd_string host, xsd_string name, xsd_string que_name, xsd_string job_id, xsd_string *result); This method returns the information about registered local resource management systems (lrms) grouped by hostname. If a type of lrms, a queue name and/or a job id have been specified, the method returns the information about specified lrms grouped by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

host: the host where the lrms to be searched is running on.
name: the type of lrms to be searched.
que_name: the name of the queue to be searched.
job_id: the identification number of the job to be searched.

OUT parameters:

result: string containing searched information stored in the local iStore db.

4.8 CPU

-int igrid_lookup_cpu(xsd_int mhz, xsd_int cache, xsd_int number, xsd_int load, xsd_string *result)

This method returns static and dynamic information about CPUs stored on the local iStore server grouped by hostname. If any input parameters have been specified, the method returns the information about all of the CPUs matching parameters values. It is possible to search the information about one or more CPUs specifying the CPU mhz, cache size, number and load parameters or a combination of these and grouping retrieved information by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

mhz: mhz of the CPU to be searched.
cache: cache size of the CPU to be searched.
number: number of the CPU to be searched.
load: load of the CPU to be searched.

OUT parameters:

result: string containing searched information stored in the local iStore database.

4.9 Memory

-int igrid_lookup_memory(xsd_int tot_ram, xsd_int tot_swap, xsd_int free_ram, xsd_int free_swap, xsd_string *result)

This method returns static and dynamic information about memory stored on the local iStore server grouped by hostname. If any input parameters have been specified, the method returns the information matching parameters values. It is possible to search the information about one or more memories specifying the total ram, total swap, free ram and free swap or a combination of these and grouping retrieved information by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

tot_ram: total ram to be searched.
tot_swap: tot swap to be searched.
free_ram: free ram to be searched.

free_swap: free swap to be searched.

OUT parameters:

result: string containing searched information stored in the local iStore database.

4.10 Network

-int igrid_lookup_network(xsd_string address, xsd_string *result)

This method returns information about network stored on the local iStore server grouped by hostname. If address parameter has been specified, the method returns the information matching its value.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

address: IP address of the host to be searched.

OUT parameters:

result: string containing searched information stored in the local iStore database.

4.11 Operating System

-int igrid_lookup_system(xsd_string name, xsd_string release, xsd_string *result)

This method returns operating system information stored on the local iStore server. If any input parameters have been specified, the method returns the information matching parameters values. It is possible to search the information about operating system specifying its name and/or the release and grouping retrieved information by hostname.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: name of the operating system to be searched.

release: release of the operating system to be searched.

OUT parameters:

result: string containing searched information stored in the local iStore database.

4.12 Device

-int igrid_lookup_device (xsd_string space, struct dimeData **dime_data)

This method returns static and dynamic information about devices stored on the local iStore server grouped by hostname. If space input parameter has been specified, the method returns

the information matching its value.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

space: device space to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore database.

4.13 Certification Authority

-int igrd_lookup_ca(xsd_string name, struct dimeData **dime_data)

This method returns the information for an accepted Certification Authority grouped by hostname or subject. If a subject has been specified, the method returns the information about specified Certification Authority grouped by hostname or Certification Authority.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the subject of the Certification Authority to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore database.

4.14 User

-int igrd_lookup_user(xsd_string name, struct dimeData **dime_data)

This method returns all of the information about registered users grouped by distinguished name. If a distinguished name has been specified, the method returns the information about accounts mapped on it grouped by hostname or distinguished name.

Returns: SOAP_OK on success, SOAP_FAULT on error.

IN parameters:

name: the distinguished name mapped on the user to be searched.

OUT parameters:

dime_data: dimeData structure containing searched information stored in the local iStore database.