



# Distributed Database Access and Data Stream Management

## Data Stream Management—Deployment Report<sup>1</sup>

Deliverable	D4.5
Authors	Tobias Scholl
Editors	Tobias Scholl
Date	28.08.2007
Document Version	1.0.0
Current Version	1.0.0
Previous Versions	0.1.0, 0.2.0

### **A: Status of this Document**

Deliverable D5 of working group 4.

### **B: Reference to project plan**

Fifth deliverable of working group *Distributed Database Access and Data Stream Management*.

---

<sup>1</sup>This work is part of the AstroGrid-D project and D-Grid. The project is funded by the German Federal Ministry of Education and Research (BMBF).

**C: Abstract**

This deliverable summarizes the changes and developments within the data stream management system from Version 0.1 to Version 0.2

**D: Change History**

<b>Version</b>	<b>Date</b>	<b>Name</b>	<b>Brief summary</b>
0.1.0	17.07.2007	Tobias Scholl	Initial draft.
0.2.0	14.08.2007	Tobias Scholl	Added paragraph on how to verify data stream services are installed without having access to the server logs. Further refinements.
1.0.0	28.08.2007	Tobias Scholl	First release.

E:

## Contents

<b>Abstract</b>	<b>2</b>
<b>Change History</b>	<b>3</b>
<b>1 Data Stream Management Installation</b>	<b>5</b>
1.1 Requirements . . . . .	5
1.2 Installation . . . . .	5
1.3 Verification . . . . .	5
1.4 Configuration . . . . .	6
1.5 De-Installation . . . . .	6
<b>2 Implementation Report</b>	<b>7</b>
2.1 Firewall support . . . . .	7
2.2 New Constants . . . . .	7
2.3 Peer Discovery . . . . .	8
2.4 Documentation, Bugfixes, new Libraries . . . . .	8
<b>3 Data Stream Management Deployment</b>	<b>8</b>
<b>4 Acknowledgements</b>	<b>9</b>
<b>References</b>	<b>10</b>

# 1 Data Stream Management Installation

This section describes the new installation procedure for the AstroGrid-D data stream management (DSM) and updates the installation procedure given in [1], which was based on Globus Toolkit 3.2.1.

## 1.1 Requirements

- Java (Version 1.5 or higher)
- Globus Toolkit (GT4.0.x). DSM only uses functionality provided by the Java WS Core of the Globus Toolkit. The Java WS Core is contained in the full installer.

## 1.2 Installation

- Extract the streamglobe.gar file from the DSM archive. If you saved the DSM archive into INSTALL\_DIR (e. g., /home/globus/gars), streamglobe.gar will be extracted into INSTALL\_DIR.  
`unzip dsm-<version>.zip`
- Deploy the data stream management services from the streamglobe.gar file. In the INSTALL\_DIR directory the command is  
`$GLOBUS_LOCATION/bin/globus-deploy-gar streamglobe.gar`  
Note: Please use the account which is used to run the container (e. g., globus). Otherwise you can run into permission issues during start-up or undeployment.
- (Re-)start the Globus container.

## 1.3 Verification

If the AstroGrid-D Data Stream Management is installed correctly, you should see the following five services running in your container:

```
.../wsrf/services/streamglobe/ContentProvider
.../wsrf/services/streamglobe/ContentProviderFactory
.../wsrf/services/streamglobe/Peer
.../wsrf/services/streamglobe/PeerFactory
.../wsrf/services/streamglobe/SpeakerPeer
```

This information is either shown on your terminal (if you start the container with `globus-start-container`) or in the log-file of your container<sup>2</sup> when you used

<sup>2</sup>e. g., `$GLOBUS_LOCATION/var/container.log`

`globus-start-container-detached` or the `start-stop` script from the Quick-start Guide.

If users want to verify, that the data stream management runs in a Globus container, they can use the general `wsrf-query` client, to query the `ContainerRegistryService`. All services running at a server are registered with this registry. The following query requests all services whose address contains the “streamglobe” term.

```
wsrf-query -a \  
-s https://buran.aei.mpg.de:8443/wsrf/services/ContainerRegistryService \  
"/**/*/*[local-name() = 'Address' and contains(text(), 'streamglobe')]"
```

If all data stream management services are installed, the query above will return content comparable to the following output:

```
<ns6:Address xmlns:ns6="http://schemas.xmlsoap.org/ws/2004/03/addressing">  
https://buran.aei.mpg.de:8443/wsrf/services/streamglobe/PeerFactory  
</ns6:Address>  
<ns6:Address xmlns:ns6="http://schemas.xmlsoap.org/ws/2004/03/addressing">  
https://buran.aei.mpg.de:8443/wsrf/services/streamglobe/Peer  
</ns6:Address>  
<ns6:Address xmlns:ns6="http://schemas.xmlsoap.org/ws/2004/03/addressing">  
https://buran.aei.mpg.de:8443/wsrf/services/streamglobe/ContentProvider  
</ns6:Address>  
<ns6:Address xmlns:ns6="http://schemas.xmlsoap.org/ws/2004/03/addressing">  
https://buran.aei.mpg.de:8443/wsrf/services/streamglobe/ContentProviderFactory  
</ns6:Address>  
<ns6:Address xmlns:ns6="http://schemas.xmlsoap.org/ws/2004/03/addressing">  
https://buran.aei.mpg.de:8443/wsrf/services/streamglobe/SpeakerPeer  
</ns6:Address>
```

## 1.4 Configuration

`$GLOBUS_LOCATION/etc/streamglobe/jndi-config.xml` is the configuration file for the data stream management. It contains the configuration for all services of the data stream management. The default configuration should be fine for most setups. Find details for some of the configuration parameters for the **SpeakerPeer** (`<service name="streamglobe/SpeakerPeer" />`) service below.

**gridDiscovery** if set to true, the `SpeakerPeer` uses grid services to verify that connected peers are alive. This is the default. Otherwise it uses multicast to do so (which of course is restricted to local area setups).

## 1.5 De-Installation

1. Undeploy the data stream management from your Globus container.  
`$GLOBUS_LOCATION/bin/globus-undeploy-gar streamglobe`

2. (Re-)start the Globus container.

## 2 Implementation Report

During the development and use of the data stream management several enhancements and bugfixes were found. These are documented in the following.

### 2.1 Firewall support

The Globus Toolkit offers the possibility to configure open ports available for outgoing and incoming connections. This enables administrators to configure their firewalls only with a limited number of open ports. This open port range is defined with the environment variable `GLOBUS_TCP_PORT_RANGE`.

If this environment variable is set during start-up of the Globus container, the data stream management picks a random port from the `GLOBUS_TCP_PORT_RANGE` for outgoing data streams and thus adheres to firewall policies. `GLOBUS_TCP_PORT_RANGE` is shared by all Grid services running in the container, and therefore, ports may be already used by an other data stream or other service. Depending on the number of available ports in `GLOBUS_TCP_PORT_RANGE`, the data stream management retries several times to open a new random port within the valid range until. Using the default port range `GLOBUS_TCP_PORT_RANGE=20000,25000`, will try 500-times (10%) until it gives up and fails.

### 2.2 New Constants

Several internal classes of our data stream management system generate character output either for data streams, logfiles and other various tasks. As also developers of user-defined operators [2] might be interested in using the default locations for writing logfiles, we describe these constants.

For example, in order to avoid having several definitions for the “new line” character, one single variable in the `streamglob.util.Constants` class is now available. This can be used for further developments and thus linebreak handling is unified across the code.

Furthermore, accessing directories within the `GLOBUS_LOCATION` is necessary in several occasions, be it in order to access third-party libraries or in order to access information specific to the data stream management, such as configuration files or directories for output. The following constants provide access to such locations:

**CONFIGURATION\_DIRECTORY** This directory contains all configuration information for the data stream management.

**OUTPUT\_DIRECTORY** Output for monitoring and logging by Globus is written to `GLOBUS_LOCATION/var`. `OUTPUT_DIRECTORY` is a `streamlobe` directory

in that location, so that generated output is separated from configuration or basic libraries. This allows for easier maintenance.

**GLOBUS\_LIBRARIES** This is a reference to the `GLOBUS_LOCATION/lib` directory. This enables us to access jar-files of other deployed services.

**GLOBUS\_LOCATION** As the start-up scripts of Globus define this variable as System property, it is very convenient to use these properties instead of being dependent on the subtle differences of operating systems when specifying environment variables. However, to keep control to what extent directories are accessible for classes of the data stream management, this constant is currently private on purpose.

## 2.3 Peer Discovery

With the current release (0.2.0), the data stream management system changed its default behaviour of speaker peers to discover other peers participating in the data stream management. It now uses Grid methods, by checking regularly the availability of the peer service at the endpoint-reference registered at the speaker peer.

## 2.4 Documentation, Bugfixes, new Libraries

The code base of the data stream management has been further improved by fixing several bugs and by enhancing the documentation of several packages. In several occasions, Generics have been introduced for better type safety and we switched to Version 0.7.0 of the JGraphT library.

# 3 Data Stream Management Deployment

The data stream management has been successfully deployed at nodes of all available AstroGrid-D sites<sup>3</sup> and also at the Leibniz Rechenzentrum (LRZ) as one of the D-Grid sites:

- AEI (buran, supergrid).aei.mpg.de
- AIP (photon).aip.de
- ARI (alnitak, hydra).ari.uni-heidelberg.de
- LRZ (lx64ia2).lrz-muenchen.de
- MPE (gavosrv1).mpe.mpg.de

<sup>3</sup>The Max-Planck-Institut für Astrophysik (MPA) currently does not provide resources.

- TUM (bladekemper16,bladekemper19,bladekemper21,bladekemper24).informatik.tu-muenchen.de
- ZIB (mardschana).zib.de

## 4 Acknowledgements

Thanks to all AstroGrid-D and D-Grid administrators for providing their valuable feedback on the installation process of the data stream management.

## F: References / Bibliography

### References

- [1] A. Carlson and T. Scholl. Distributed Database Access and Data Stream Management: Prototype for Manual Query Execution Plans. Deliverable D4.2, AstroGrid-D project, August 2006. <http://www.gac-grid.de/project-documents/deliverables/wp4/wg4-d2-1.0.0.pdf>.
- [2] T. Scholl and A. Reiser. Distributed Database Access and Data Stream Management: Distributed Function Providers. Deliverable D4.4, AstroGrid-D project, February 2007. <http://www.gac-grid.de/project-documents/deliverables/wp4/wg4-d4-1.0.0.pdf>.