

GABBs Installation Instructions

This document describes how individual GABBs components can be added to a pre-existing hub. For detailed instructions on how to get started with a HUBzero hub on your own infrastructure, refer to <https://hubzero.org/documentation/2.1.0/installationrh>. Please note that this document assumes that you have a HUBzero 2.1.x hub running on either a RedHat or CentOS machine. Debian-based hubs are not supported. All of the software packages (RPMs) mentioned in these instructions are hosted at <http://packages.hubzero.org/>. Please note, that you will need to add this package repository to your server before installing any of these packages. Instructions for adding the GABBs package repository can be found below. Also note that copy-pasting the commands directly from this document may not work; you may need to type them out explicitly.

GABBs is made up of several inter-related but independent software building blocks. Each of these can be installed and work independent of the others. There are four primary building blocks:

1. A geospatial data management infrastructure, iData that includes web-based file management, special handling including automatic metadata extraction and preview support and map-enabled keyword search for geospatial files.
2. MultiSpec, an image visualization and analysis tool for multi and hyperspectral satellite data.
3. GeoBuilder, a visualization tool for building complex views combining geospatial and time-series data with no programming.
4. pyMapLib, a Python map library for programmatically constructing map viewer widgets and geospatial file overlays.

iData Installation Instructions

The addition of iData to an existing hub involves the setup and configuration of additional servers and software. iData implements a storage provider for HUBzero project files with special value-addition for geospatial data. The physical geospatial files uploaded to iData are managed by a separate server running the iRODS distributed data management system. The geospatial value addition for these managed files includes three specific enhancements:

1. Metadata such as geospatial bounds, projection information and feature descriptions are extracted automatically on file upload and indexed for subsequent search.
2. The iData web interface supports a quick preview and overlays of various geospatial file formats.
3. iData file support is integrated into the primary hub search interface, enabling both keyword-based search and geographic filtering using user-defined bounding boxes.

These three enhancements are implemented via iRODS microservices that enable both automatic and on-demand rule-based triggers that run arbitrary code in response to various file events. In this case, metadata is extracted via a trigger that runs on file upload, user requests for a file preview trigger an on-demand rule that processes and registers a map layer to a local GeoServer map server and finally, metadata extracted on file upload is automatically indexed to Apache Solr running on the hub's webserver. These microservices are designed for iRODS version 4.1.x, but work best with version 4.1.8. Generic instructions for setting up an iRODS server on a CentOS machine can be found on the iRODS downloads page (<https://irods.org/download/>). However, we recommend installation using our GABBs-specific iRODS packages that simplifies the setup and configuration of the iRODS server and our microservices.

Step 1: Add GABBs package repository

We will install iRODS, the associated microservices and the GeoServer map server on a separate server. Start a new CentOS or RedHat 6.x machine and add the GABBs package repository to it:

Download the package repository RPM here:

<https://packages.hubzero.org/rpm/rhel6/gabbs/gabbs-repo-0-1.el6.noarch.rpm>

At the shell prompt type:

- `sudo rpm -i gabbs-repo-0-1.el6.noarch.rpm`

This will add the GABBs package repository to the yum.repos list and allow us to now use yum to install the remaining packages.

Step 2: Install iRODS dependencies

iRODS relies on a PostgreSQL server for managing file metadata. The various dependencies and ODBC APIs are combined into a single dependency package which can be installed with the following command (as root):

- `yum -y install irods-dep`

Step 3: Install iRODS and the various microservices

The setup and configuration of iRODS and the GABBs microservices are folded into a single package installation with dependencies. Next run (as root):

- `yum -y install gabbs-microservices`

This step may take some time depending on your network speeds since it includes the installation of several package dependencies and configuration tasks.

Step 4: Install GeoServer mapserver

Since GABBs relies on a specific GeoServer version, we package that version along with other configuration scripts and style files in a single package. To install this package, run (as root):

- `yum -y install gabbs-geoserver`

Step 5: Configure GeoServer information in iRODS

Since the GABBs preview microservices need to register map layers to GeoServer, iRODS needs to be aware of the GeoServer connection information. We will set this up as a server configuration parameter in iRODS. At this point, you have the option to either use the default GeoServer admin password of “geoserver” or provide an alternate string, which would be set to be the new GeoServer admin password. Run as root:

- `/tmp/set-geoserver-password.sh <geoserver admin password>`

Step 6: Install GeoServer styles

One of the kinds of geospatial files that can be previewed in iData are vector files that can contain various geometric shapes such as points, lines and polygons. Since iData allows several files to be overlaid on the same map, a color map is applied to distinguish the geometric shapes of one layer from the other. These color maps are defined as map styles in GeoServer so that they can be applied to the layer when served from the map server. In order to install these pre-defined styles, run as root:

- `/var/lib/irods/geoserver/styles/install-styles.sh <geoserver admin password>`

Step 7: Create non-admin iRODS user

The iRODS installation in [Step 3](#) only sets up an admin iRODS user that can manage the iRODS server and resources. iRODS admin users are typically not used to manage physical files, hence we set up a new non-admin iRODS user to manage the geospatial files in iData. Here, you have the option to choose your own iRODS username and password. As root, run:

- `/tmp/add-irods-user.sh <irods username> <irods user password>`

Step 8: Set up daemons

As a final installation step, we setup several daemons that ensure that our services (iRODS server, IP tables, GeoServer) are restarted automatically in the correct order if the machine is rebooted. As root, run:

- `yum -y install gabbs-irods-services`

At this point, we have a separate CentOS or RedHat server that contains the iRODS server, the GABBs microservices and a GeoServer map server that provide the physical file storage, metadata extraction and preview capabilities of iData. Next, we will add the iData web file management interface to the hub's project files area and setup the storage provider backed by this iRODS server.

Step 9: Add GABBs package repository to hub webserver

We will again use our hosted packages to install and configure the iData storage provider on the hub webserver. As a first step, add the GABBs package repository to the hub webserver. As root on the hub machine, download the RPM that sets up the GABBs repository at: <https://packages.hubzero.org/rpm/rhel6/gabbs/gabbs-repo-0-1.el6.noarch.rpm>. Next, install this RPM:

- `rpm -i gabbs-repo-0-1.el6.noarch.rpm`

Step 10: Install iRODS client packages

The iData web interface manages the physical files stored in iRODS via a FUSE mount on the hub webserver. This allows the iRODS files to be accessible just like a file on the hub webserver's local filesystem, simplifying both read and write access. iRODS provides a client package that include the FUSE client. As a first step, we install this package as root:

- `yum -y install irods-client`

Step 11: Setup connection to iRODS server

In order to set up the FUSE mount to be accessible by the hub's web user; we first need to be able to connect to the iRODS server from the hub webserver acting as a client. While, the previous step installed the necessary remote client commands, we need to store the iRODS server's DNS and username details. As root, run:

- `/usr/bin/update-irods-connection.sh <iRODS server's DNS> <irods username>`

Note, that the “irods username” above, is the non-admin user we set up on the iRODS server in [Step 7](#).

Step 12: Install iData web interface and related dependencies

We now install the iData web interface (implemented as a set of HUBzero plugins, template overrides and database migrations), dependencies and software for managing the iRODS FUSE mount. As root, type:

- `yum -y install gabbs`

Step 13: Install clamav virus scanner

All files uploaded to iData need to be scanned for viruses. We use the clamav antivirus software. As root, run:

- `yum -y install clamd`

Step 14: Create configuration file

Rather than repeatedly type the iRODS username and password when performing various configurations, we instead save these details to a configuration file that can be parsed by the various scripts. As root, create the file “/etc/gabbs.secrets” on the hub web server using your favorite editor with the following contents:

```
GABBSUSER=<irods username>  
GABBSPASSWD=<irods user password>
```

Note, that the “irods username” and “irods user password” here were set up in [Step 7](#).

Step 15: FUSE mount setup

Next, we setup the FUSE mount to access files from the iRODS server. As root, run:

- `service irods-fuse start`

Step 16: GABBs plugin installation

Next, we configure and install the various GABBs plugins implementing the iData web interface. As root, run:

- `hzirods install hub <irods server DNS> <irods username> <irods user password>`

Step 17: Setup hub search for iData files

iData files are now integrated into the hub search mechanism, allowing for both keyword-based and map-based search for iData files. Here, we install the necessary template overrides to add the map-based search interface and inform the iRODS server of the hub's Solr endpoint to index file metadata to. As root, type:

➤ `yum -y install gabbs-search`

followed by:

➤ `/usr/bin/run-irods-rule.sh <hub webserver DNS> <irods user password>`

Note, that this command uses the hub webserver's DNS as an argument. There are two other important points to consider when running these commands:

1. The view overrides adding map-based search are installed to a hub template named "mytemplate". If your hub installation uses a different template; you will need to copy these view overrides to that template. Essentially, copy `<docroot>/app/templates/mytemplate/html/com_search` over to your desired template at the same relative file path.
2. The shell script setting the Solr endpoint in the iRODS server's configuration assumes that the Solr port is the default 8445. Contact us if you would like to use a different port number.

Step 18: Set up daemons

As in [Step 8](#), we will now setup some daemons on the hub web server to automatically restart services (including the FUSE mount) on server reboots. As root, run:

➤ `yum -y install gabbs-hub-services`

followed by:

➤ `service iptables save`

This latter step is required in order to save all the firewall modifications that have been made to allow access from the iRODS server.

As this point, you should have a working iData installation backed by an iRODS server.

MultiSpec Installation Instructions

Installing the MultiSpec tool is straightforward since it is bundled with all of its dependencies in a single package. To add MultiSpec to your hub, run as root on the hub webserver:

➤ `yum -y install gabbs-multispec`

Note: You will need to first add the GABBs package repository to your hub web server before attempting to install MultiSpec. If you haven't already done so, follow the instructions on [Step 9](#).

GeoBuilder Installation Instructions

The GeoBuilder tool uses a combination of hardware and software-based rendering of geospatial files and map layers. Hence, it relies on an external Debian-based (7 or 8) GPU machine that functions as the render server. Contact us for details on setting up the render server. Once the render server has been set up, GeoBuilder can be installed as follows. As root, run the following on the hub web server:

➤ `yum -y install gabbs-geobuilder`

followed by:

➤ `/usr/bin/set-geovis-server.sh <IP address of render server>`

Two things are of important note here:

1. The GeoBuilder installation will setup a separate Rappture library that contains the new map elements used for the GeoBuilder tool interface. Please ensure that the hub web server has at-least 8 GB of free disk space.
2. The “set-geovis-server.sh” script above modifies the hub middleware to add details of the render server to be used for GeoBuilder tool sessions. If you would like to save any custom modifications to the middleware configuration at /etc/mw-service or /etc/mw-client, then ensure that you save a copy before installing the gabbs-geobuilder package and running this script.

pyMapLib Installation Instructions

The pyMapLib library is installed as a shared library in /apps on the hub web server and can be imported into any tool session via “use gbsmaps-22”. A demonstration tool for this package is also included with the installation. To install pyMapLib, run as root:

➤ `yum -y install gabbs-pymaplib`