Data Repositories

Data Repositories are locations where Ergo looks for and stores all of the data that it uses and creates. A Repository can be located locally, as a file on the local machine drive, or be remote. By default, Ergo creates a local cache repository on the user's system, where any remote data that is accessed is cached for local use.

Local Repository

Local repositories are simply folders on the local machine's drive, which are formatted in a particular format so that Ergo recognizes and knows how to read /write them. You can create a new local repository by using the File->New->Repository function in Ergo. Once you select "local repository" and select a drive location, Ergo will set up the appropriate folder structure and initialize the repository. This is useful for creating and editing test data run locally. It is recommended to let Ergo manage the folder and file structure of the repository. There are implicit links between certain files, and certain formats that are expected. Modifying it by hand might corrupt the repository if you don't have an understanding of how the repository format works.

Remote Repositories

To share or publish data to other Ergo users, you will want to use a Remote Repository. These are servers that can be connected to by multiple Ergo users.

WebDAV repository

The most common type of remote repository is a WebDAV repository. To connect to a WebDAV repository, you simply need to enter the WebDAV url that is provided by the repository owner. If necessary, you can enter a username/password if it is password protected. The most common case for ERGO WebDAV repositories is that the data is publicly accessible for reading without a password, but a username/password must be supplied for writing to the repository.

To create a WebDAV repository, you will need to set up a WebDAV server, which is beyond the scope of this article. Once it is set up, you can provide a URL to the webDAV root, or a subfolder in webDAV, where you'd like the repository to live. (Please note that if you allow un-authenticated write access, you expose yourself to possible abuse from 3rd parties on the internet.) Because you can set up subfolders as their own Ergo repository, it is possible to have multiple different Ergo repositories on a single WebDAV server.

PostGIS Repository

You may also connect to special repositories that are hosted by Posgresql databases with the PostGIS extension. This allows for some advanced behavior, just as geospacial querying. Setting up a postGIS repository is a relatively complicated procedure, and beyond the scope of this document.

Creating a new repository type

To create a new repository type, you must extend the edu.illinois.ncsa.ergo.gis.repositoryTypes extension point (which is defined in the edu.illinois.ncsa.ergo.gis plugin). You must specify an id and tag, and a class that implements the GISDatasetRepository interface (in the package edu.illinois.ncsa.ergo.gis). To make this easier, Ergo provides an abstract class, edu.illinois.ncsa.ergo.gis.repositories.BaseRepository, which implements many of the basic methods.

Implementing a new repository is fairly complex, it is recommended that you study the source code for LocalRepository and SamRepository (which is the class that implements the WebDAV repository type).

Default repository directory layout

While it is generally recommended to not modify a repository's data by hand, understanding the layout of the Local and WebDAV repository directories will help a developer debug issues that arise while implementing new features.

The repository folder contains the following sub-folders. Each is organized hierarchically based on dataset schema ids.

- properties – this folder contains an xml file in a custom format that defines meta-data about each dataset, including, but not limited to:
    - location – a unique URL that identifies the dataset. Datasets that were originally from a different repositories can include the original URL so they can provide a copy of the original dataset
    - description – a tag to describe the location ID. This is used to identify matching datasets across different repositories
    - maeviz-mapping – a set of mappings that map between the dataset's field names and the field names that Ergo expects
- datasets – this folder contains the actual data files for each dataset, grouped in folders by dataset schema ids.