



---

## iRODS Deployment

### About

Technical documentation about iRODS deployment.

**Modified:** 29 January 2018

### Synopsis

This document discusses the basics of deploying and configuring iRODS, the software EUDAT uses so as to effect safe replication (B2SAFE). Another two actions are required, [acquire a Handle prefix](#) and [configure safe replication](#). This document is dedicated to iRODS.

### Introduction to iRODS

[iRODS](#) is a community-driven, open-source, data grid software solution. It provides a means for managing large distributed collection of digital objects, maintaining metadata and applying data management policies. iRODS comes with a comprehensive set of generic capabilities, although in EUDAT practice only a subset is required. The functionality can be further extended by user-defined rules and micro-services, written in an iRODS-specific language, or by implementing new modules. Hence, it can be tailored exactly to the end-user and community needs. This document does not intend to describe all features of iRODS but rather aims at providing the basic information needed to make a decision regarding deployment and installation options. For more information we refer the reader to iRODS documentation and related work section.

Central component of each iRODS installation is the metadata store iCAT. It is a SQL database managed by iRODS, which contains both information needed for authentication and authorization decisions as well as user and system metadata describing data objects managed by iRODS. It is possible to use an existing SQL database or to use the all-in-one iRODS bundle which comes with its own PostgreSQL database.

iRODS uses the notion of an abstract storage resource i.e. a software/hardware system able to store data. Examples of storage resource types supported by iRODS are Unix file system, HPSS and Amazon S3. But it is possible (and quite easy) to extend this list by providing an implementation of the resource adapter interface. The standard installation of iRODS creates the initial resource (so called Vault) on which the data can be ingested.

Data managed in iRODS are presented as hierarchical collections of objects. The data are physically stored on the storage resources. Users use logical names of collections of objects to manipulate their content. Example of a logical name would be:

```
/tempZone/home/testuser1/file.txt
```

The logical name is composed of the zone name (tempZone), path (/home/testuser1) and actual object name (file.txt). A data object has only one logical name but can be replicated on multiple storage resources, thus it can have a number of physical locations.

A more complex use case is to use iRODS to access existing data sets. This can be achieved by using the mounted collections. iRODS does not maintain any metadata information for such collections, in particular it has no information about subdirectories and files in the collection. When using this option, iRODS works as a "proxy": each time a request for a data object from a mounted collection is issued, the request is simply passed to the

underlying file system. The advantages are twofold. Already existing data can be easily and quickly made available via iRODS (without labor-intensive and time-consuming re-ingest). Secondly, it is possible to modify the content of the mounted collections by using tools other than iRODS, without the danger of creating inconsistencies. For a normal collection, any low-level direct access and modification (e.g. physical removal of a file on the resource) would lead to an inconsistency, as iRODS would keep the record describing the data object (file) in the iCAT database although the object will not be present anymore. The major disadvantage of the mounted collection is the fact that the metadata functionality is not available.

It is possible to create a group of iRODS servers managing distributed resources. A group of such servers is called iRODS Zone. It is important to stress that a zone always includes exactly one iCAT server. Thus, a zone usually represents a single administrative domain. iRODS also provides a means of connecting distinct administrative domains to create an iRODS Federation. As explained above, each of the zones maintains its own user database and information needed for authorization decisions, in its iCAT. When an access request to local resources from a remote zone is issued, iRODS delegates the authentication request to the home zone of the requesting user. Upon a successful authentication acknowledged by the home iRODS server, an authorization decision is made based on a locally available information for given remote user. Therefore it is possible (and necessary) to define local authorization policies for each user from the remote zone.

The replication of a data object in iRODS means physical copying to a new resource within the same zone, in such a fashion that is transparent to the user (i.e. the logical name does not change). Upon access to a particular data object, iRODS can serve its content from one of the replicas (for instance from a physical resource which is closest to the requesting user). The replication as defined in B2SAFE refers to a process of moving files between federated zones.

## iRODS installation

iRODS is written in C and sources and packages can be downloaded from the [project download page](#) as a tarball containing all necessary files. iRODS v4.2.X can be installed by the general unix package managers e.g. apt or yum.

EUDAT has standardised on v4.X of iRODS in order to incorporate security fixes and take advantage of updated features.

iRODS does not have to run as a root service. You can and should create a separate user for the iRODS server (user in the operating system is meant here, not to confuse with the initial admin user name for iRODS).
---

Please refer to the most up-to-date documentation about iRODS listed in <https://docs.irods.org>.

## Support

Support for iRODS deployment is available via the EUDAT ticketing system through the [webform](#).

If you have comments on this page, please submit them through the [EUDAT ticketing system](#).

## Document Data

**DocVersion:** 2.1

**Authors:**

Jedrzej Rybicki, [j.rybicki@fz-juelich.de](mailto:j.rybicki@fz-juelich.de)



Benedikt von St. Vieth, [b.von.st.vieth@fz-juelich.de](mailto:b.von.st.vieth@fz-juelich.de)

**Editors:**

Kostas Kavoussanakis, [kavousan@epcc.ed.ac.uk](mailto:kavousan@epcc.ed.ac.uk)

Albert Heyrovsky, [a.heyrovsky@epcc.ed.ac.uk](mailto:a.heyrovsky@epcc.ed.ac.uk)

Roger Downing, [roger.downing@stfc.ac.uk](mailto:roger.downing@stfc.ac.uk)

Christine Staiger, [christine.staiger@surfsara.nl](mailto:christine.staiger@surfsara.nl)

[Read more](#)