iRODS

TECHNICAL OVERVIEW

Imagine

you were responsible for petabytes of genome sequencing data, which could underpin decades of medical breakthroughs? Or the digitized cultural heritage of an entire nation? Or thousands of square kilometers of geophysical data that could unlock new energy resources?

What if you had files that needed to outlast the infrastructure you have today? Files that would be impossible to reproduce if they were lost? What if you had to ensure compliance with regulatory data retention and privacy policies like HIPAA, EU Directive 95/46/EC, Sarbanes-Oxley, and other federal regulations?

You would need software that can implement your data management policies, software that stands the test of time and frees you from vendor and technology lock-in.

The Integrated Rule Oriented Data System, iRODS, is open source data management software designed to manage data sets that are big, important, and complex. Hundreds of organizations worldwide use iRODS to control their records and research data.

This Technology Overview describes the pieces of iRODS that, together, enable organizations to easily move, organize, find, process, share, and secure their data. It explores the ecosystem of plugins and software clients which further extend how iRODS manages and presents data. And it connects you to additional, current information about iRODS technology.

To manage data at scales of hundreds of petabytes, billions of files, and time periods of decades, iRODS implements four main functions:

- Data Virtualization
- Data Discovery
- Workflow Automation
- Secure Collaboration

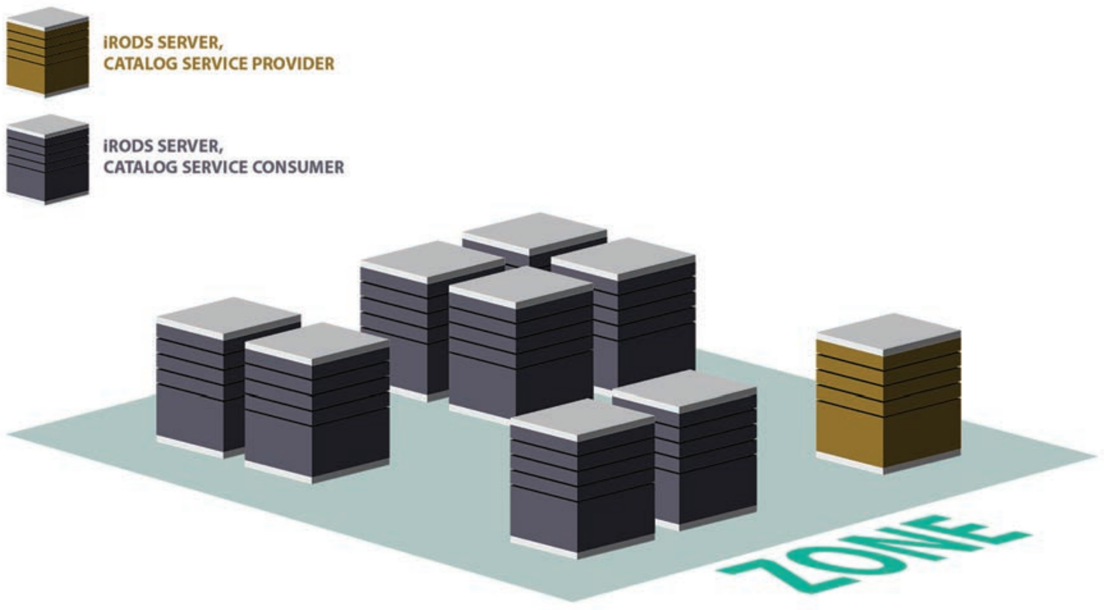
iRODS Zone

An iRODS Zone is a network of computers running the iRODS server software. Zones serve data hosted on connected storage devices, and metadata, stored in a metadata catalog.

In each Zone, one server plays the role of catalog server and connects to a relational database that holds the catalog. The other servers in the Zone are consumers of the catalog service.

All iRODS servers accept connections from iRODS clients. All iRODS servers can host Storage Resources, described in the next section, which contain the files served by a Zone. And all iRODS servers can execute iRODS Rules, which implement Workflow Automation, discussed later.

Adding servers can enhance the performance, security, and resilience of a Zone by providing redundancy, both within a single location and distributed geographically.


iRODS SERVER, CATALOG SERVICE PROVIDER
iRODS SERVER, CATALOG SERVICE CONSUMER
ZONE

Data Virtualization

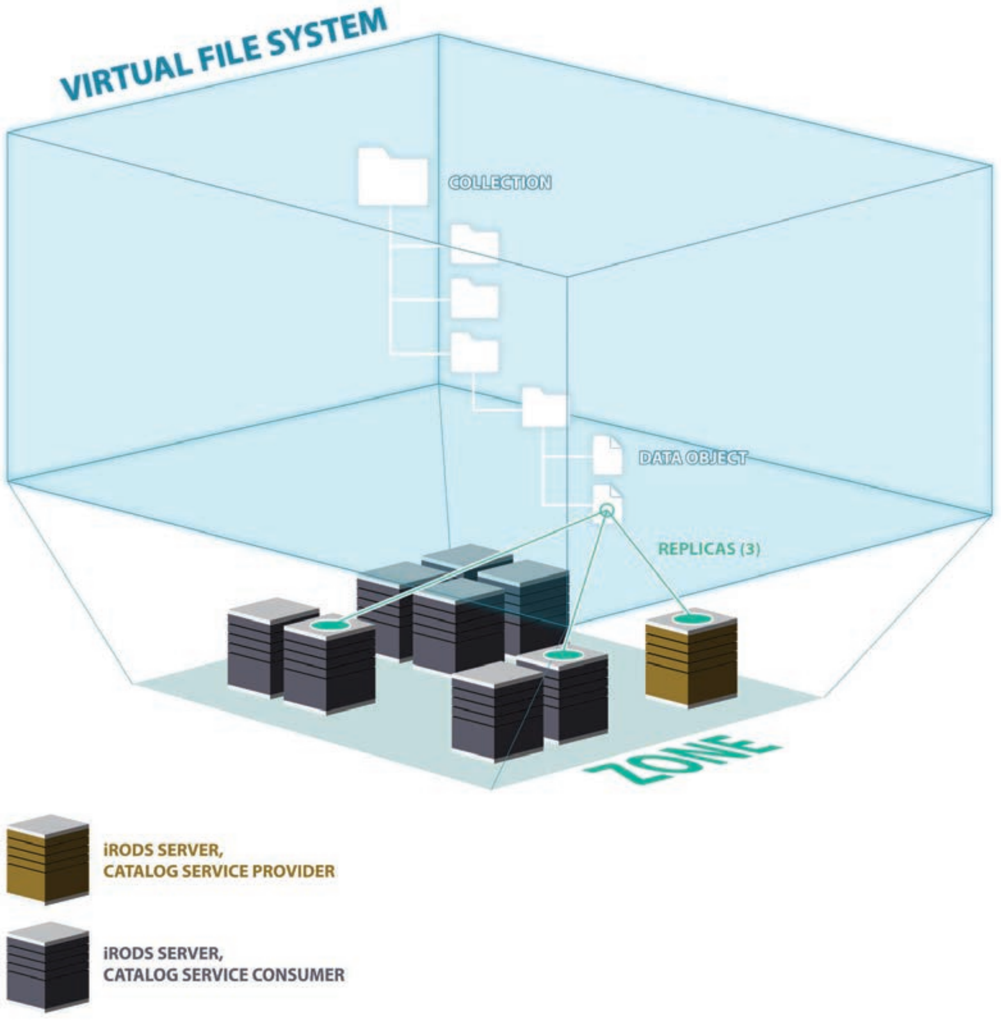
iRODS provides a logical representation of files stored in physical storage locations. We call this logical view a virtual file system and the capabilities it provides, Data Virtualization.

Virtual File System

Data stored in iRODS is typically accessed through an iRODS client. iRODS clients present files as Data Objects organized into Collections. For the most part, there is little difference between Data Objects and files, and between Collections and subdirectories. However, there are a couple important distinctions:

- Collections make no reference to the physical storage path. It is possible for two Data Objects in a Collection to be stored in different physical locations.
- A Data Object may refer to multiple Replicas. Replicas are exact copies of a file, located in multiple physical locations.

Data Objects and Collections are stored in Storage Resources in an iRODS Zone. Each Storage Resource has a name — the Resource's logical representation — and a hostname and path — the physical representation of the Resource, where files are kept. The hostname is the network name of the device that serves the data, and the path is the local file system path or object storage bucket that holds the data.


VIRTUAL FILE SYSTEM
COLLECTION
DATA OBJECT
REPLICAS (3)
ZONE
iRODS SERVER,
CATALOG SERVICE PROVIDER
iRODS SERVER,
CATALOG SERVICE CONSUMER

Composable Resources

iRODS Composable Resources allow data distribution policy to be defined by decision trees of Coordinating Resources and Storage Resources. Coordinating Resources, branch nodes of the decision tree, actively make decisions about which leaf node, or Storage Resource, will receive or serve each Data Object. There are many different types of Coordinating Resources, each with a defined logic that determines how Data Objects are distributed or accessed.

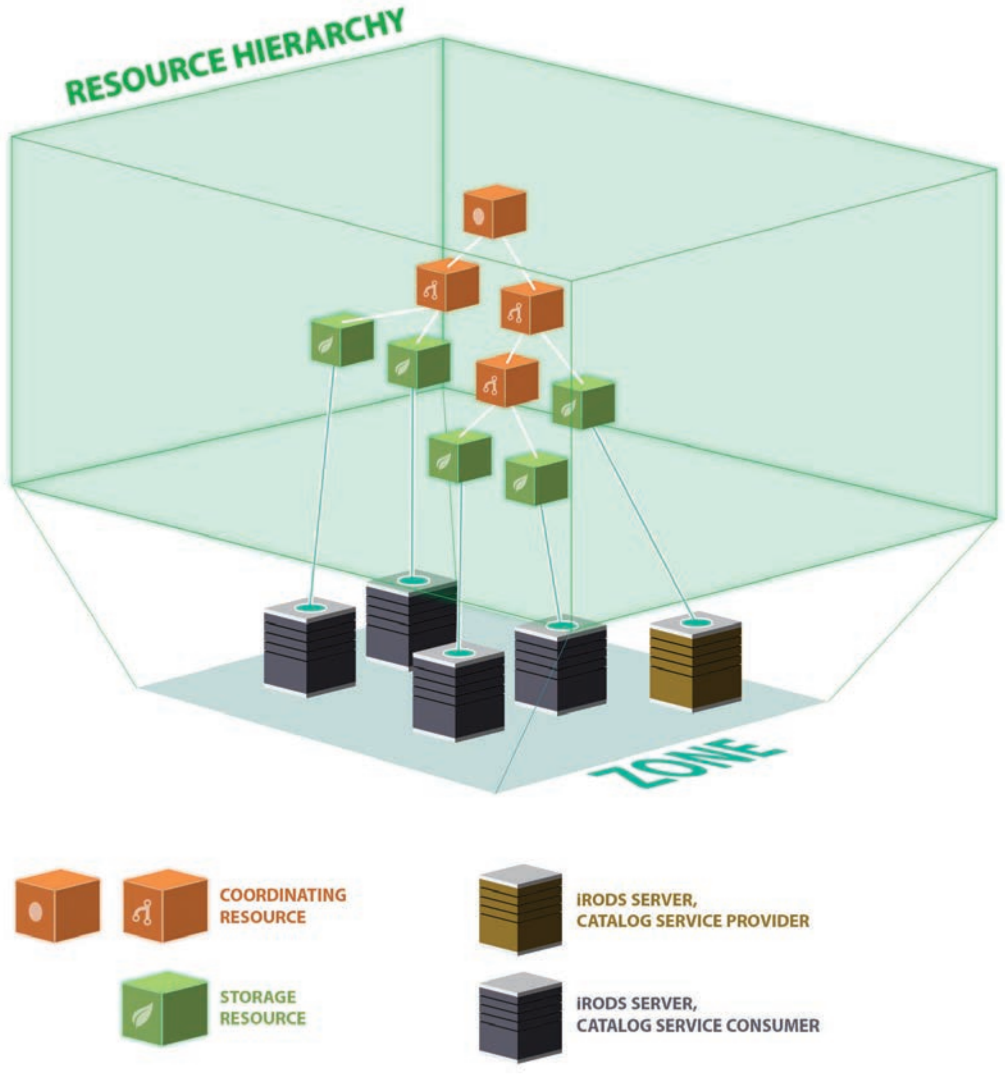
Examples of Coordinating Resource types include:

- **Random** Randomly distributes files to its Storage Resources
- **Replication** Distributes a Replica to each of its Storage Resources
- **Deferred** Distributes files based on votes determined according to an algorithm defined for each Storage Resource

For more information about iRODS Coordinating Resources, see *https://docs.irods.org/4.1.8/manual/architecture/#composable-resources*

Transferring Data

iRODS servers act as data brokers during transfers. Each server is aware of its Zone's resource composition. When an iRODS client requests a data object, the contacted server refers the request to the catalog server in order to determine the file size and location(s). The originally contacted server consults the Storage Resources identified in the catalog and uses the resource composition tree to determine which Resource will serve the file. Small files are relayed through the contacted server. For large files, the contacted server brokers a high speed parallel connection between the client and the server that hosts the most appropriate Storage Resource.

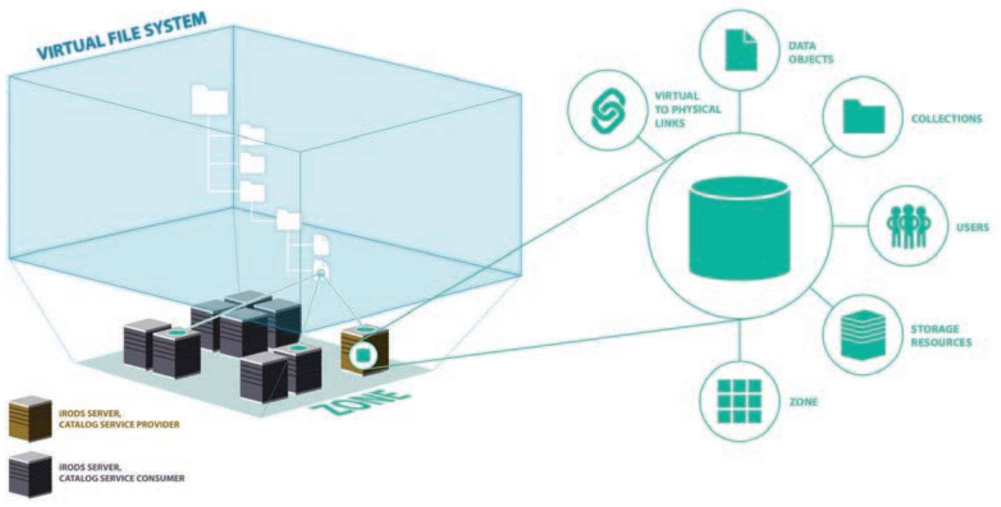

RESOURCE HIERARCHY
ZONE
COORDINATING RESOURCE
STORAGE RESOURCE
iRODS SERVER, CATALOG SERVICE PROVIDER
iRODS SERVER, CATALOG SERVICE CONSUMER

Data Discovery

The metadata catalog contains information about a Zone's Data Objects, Collections, Users, Storage Resources, as well as information about the Zone itself.

This information about data, called metadata, is extremely useful for Data Discovery, locating relevant data within large data sets. Data Object metadata includes rich, user-defined metadata in addition to traditional system metadata, such as filename, file size, and creation date. This rich metadata allows data to be identified by characteristics such as author names, keywords, case ID, and content type.

Rich metadata can include whatever descriptors you choose to apply to your data. Rich metadata can also be applied to Collections, Users, Resources, and other iRODS Zones. The entire iRODS catalog for a Zone is contained in a relational database. Currently, that database must be hosted in a PostgreSQL, MySQL, or Oracle database management system.



VIRTUAL FILE SYSTEM
DATA OBJECTS
VIRTUAL TO PHYSICAL LINKS
COLLECTIONS
USERS
STORAGE RESOURCES
ZONE
ZONE
iRODS SERVER, CATALOG SERVICE PROVIDER
iRODS SERVER, CATALOG SERVICE CONSUMER

This information about data, called metadata, is extremely useful for Data Discovery, locating relevant data within large data sets.

Workflow Automation

Once data is stored and available in the catalog, it often needs to be migrated, secured, or otherwise processed.

Each iRODS Server runs a Rule Engine that is an event-triggered background process. The Rule Engine is programmed using iRODS Rules, which specify what actions should be triggered when iRODS initiates a particular system activity.

iRODS event triggers are called Policy Enforcement Points (PEPs). Consider, for example, a rule to transfer ownership of data objects to the project manager when a user is deleted; the trigger — or PEP — is the deletion of the user. Similarly, rules could be written to extract metadata or pre-process data whenever a file is uploaded to an iRODS Resource.

Chaining rules and PEPs allows you to create powerful, customized workflows that save time and prevent human error. Complex multi-step scientific processes can be tightly managed and automated by keeping thorough records of ongoing status and other lab information, and only alerting humans when necessary. Organizational data management policy can be captured in an automated, auditable fashion using iRODS rules.

Beginning in iRODS 4.2, the iRODS Rule Engine can be extended with Plugins. With Rule Engine Plugins, rules can be defined in any programming language for which a Plugin exists, for example, Python. The example on the next page shows a Python-based rule for extracting image metadata once a file is put (uploaded) into an iRODS Zone.

The native base iRODS Rule Engine uses a domain specific programming language called the iRODS Rule Language. The Rule Language makes use of external functions called microservices — pre-packaged C++ programs designed to accomplish specific tasks. iRODS microservices have access to in-memory session variables from iRODS.

Other examples of iRODS Rule sets or policies include:

- **Tiered Storage** Files can be migrated from a high performance file system to an object store if they have not been used in 90 days.
- **Access Control** Read-write-own permissions can be assigned according to the metadata associated with a Data Object or Collection.
- **Landing Zone** Raw data can be staged to a high performance computing (HPC) system for processing through the native file system, and processed data products can be automatically ingested back into iRODS.
- **Auditing** iRODS can log every read or write to every file, or a subset of this activity, to a file or message bus.
- **Transformation** iRODS can change file formats or transform file content upon upload or movement of data.

The below rules, one written in the iRODS Rule Language and one in Python, together extract EXIF metadata from image files and add the metadata to the iRODS catalog. Note the two microservices called in the Rule Language: *msiString2KeyValPair* and *msiAssociateKeyValuePairsToObj*. These are built-in microservices and already available. Also note the name of the Python function, acPostProcForPut. This is a PEP reached immediately after a file is added, or "put" into iRODS.

iRODS rules for EXIF metadata extraction. These rules perform the following operation: 1) once a file is added to iRODS, 2) get the file path, 3) log the file path, 4) get the logical path, 5) log the logical path, 6) open each image, 7) extract the metadata, 8) add the metadata to a string, 9) deconstruct the string and apply the metadata attribute-value pairs to the iRODS Data Object, 10) log process completion.

```
# existing iRODS Rule Language - custom.re
add_metadata_to_objpath(*str, *objpath, *objtype) {
  msiString2KeyValPair(*str, *kvp);
  msiAssociateKeyValuePairsToObj(*kvp, *objpath, *objtype);
}
```

```
# Python - core.py
#################################
# EXIF Extraction Demo
#################################
import os
import sys
from PIL import Image
from PIL.ExifTags import TAGS
def acPostProcForPut(rule_args, callback):
  phypath = callback.getSessionVar('filePath', 'dummy')[1]
  callback.writeLine('serverLog', phypath)
  objpath = callback.getSessionVar('objPath', 'dummy')[1]
  callback.writeLine('serverLog', objpath)
  exiflist = []
  for (k, v) in Image.open(phypath)._getexif().iteritems():
  exifpair = '%s=%s' % (TAGS.get(k), v)
  exiflist.append(exifpair)
  exifstring = '%'.join(exiflist)
  callback.add_metadata_to_objpath(exifstring, objpath, '-d')
  callback.writeLine('serverLog', 'PYTHON - acPostProcForPut() EXIF complete')
```

Secure Collaboration

Data is most useful when it’s in the hands of the right people. There is a recognized need in the public research community to publish data sets that accompany written articles.

Even in fields where data may not be published, it is usually necessary to share data sets between multiple workgroups. However, as data sets grow beyond several gigabytes, it becomes difficult to impossible to move the data between locations. iRODS provides Secure Collaboration through three technologies: Tickets, Permissions, and Federation.

- iRODS Tickets provide controlled public access to Data Objects and Collections. The owner of a Data Object or Collection can create a Ticket and share it with non-iRODS users to grant them read or write access. Tickets can be revoked, and they can be set to automatically expire upon a specified date and time or a specified number of reads or writes.
- iRODS Permissions are analogous to UNIX file system permissions. The owner of a Data Object or Collection can assign read or write access for any number of defined iRODS Users and Groups. Group membership is defined by the administrator(s) of a Zone.
- iRODS Federation extends data sharing and publication beyond a single Zone. In a Federated deployment, once the administrators of two iRODS Zones share a set of keys, the owner of a Data Object or Collection can assign read and write permissions to users from outside Zones. When reading or writing data, the transfer mechanism is analogous to that for a single Zone. Unless the file is very small, iRODS servers broker a connection between the server containing the data and the client requesting it. As a result, Federation enables high-performance access to data stored in any other iRODS Zone.

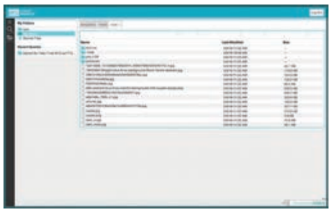
Federation solves the issues of scalability, perishability, and transfer rates inherent with a centralized sharing system. Management responsibility is distributed to individual Zones, and each Zone can provide its own view (i.e., interface) to data in any iRODS Zone in the world.

iRODS Clients and APIs

Users access the data and the metadata in iRODS through iRODS clients. Clients communicate with iRODS through an API, and some clients use a stack of multiple APIs.

The following is a list of several iRODS clients. This is not an exhaustive list, as new developments emerge quite often.

Web-based Graphical Interfaces

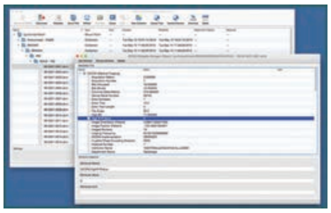


Cloud Browser
One of our founding members, the DICE Group, has released Cloud Browser. It manages iRODS collections, transferring files to and from collections, annotating files and collections with metadata, metadata search, iRODS rule executions, and more. Cloud Browser is available from DICE-UNC on GitHub.

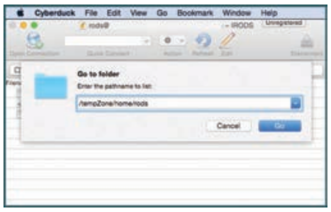
MetaLnx
Adds metadata search and administrative capabilities to iRODS file management. Administrative features include server health checks, a user account directory, and permission management. MetaLnx is available for beta testing through EMC. Contact us for a demo and to get in touch with EMC.

iRODS clients may be customized for an organization’s particular needs, and anyone is free to develop a new iRODS client, open source or otherwise.

Desktop Graphical Clients



Kanki
Qt-based iRODS client developed by Ilari Korhonen at University of Jyväskylä, Finland. Kanki is a very fast and resposive client. In addition to file and collection management, it also has advanced features like metadata templating and metadata search (in beta). Kanki is an open source project on GitHub.


Cyberduck File Edit View Go Bookmark Window Help
Go to folder
Enter the pathname to list:
/tempZone/home/rods
Cancel
Go

Cyberduck
Popular file transfer client that speaks to a host of systems, including FTP/SFTP, Amazon S3, Swift, and now iRODS. iRODS support is included thanks to the efforts of the iPlant Collaborative, the DICE Group, and the Cyberduck development team. More information is available at Cyberduck.io

Network File System Interfaces

WebDAV
Mike Conway from the DICE Group has developed a WebDAV client for iRODS, based on the Milton.io WebDAV framework. WebDAV support is built in to Windows, MacOSX, and Linux, making it possible to drag and drop files from your computer into iRODS collections. Project is available on GitHub.

FUSE
The FUSE iRODS client lets you mount your iRODS home collection as a file system on Linux or MacOSX. FUSE is available at https://irods.org/download/

PARROT
Parrot is a program that can “wrap” another program, trap file system accesses (reads/writes), and direct those accesses to iRODS or another protocol. This allows you to use iRODS as the storage backend for almost any program without re-writing, re-linking, or re-installing. Parrot is part of the Cooperative Computing Tools Suite (cctools) from the Cooperative Computing Lab at Notre Dame. The cctools suite is available on GitHub.

Command-Line Interfaces

iCommands
iCommands are command line executables that ship with iRODS. The iadmin command is used to perform a number of administrative activities. Many of the other iCommands are analogs to UNIX command line activities. iput and iget are used to add and retrieve data to and from collections; icd is used to navigate collections. imeta and iquest are used to manage and query metadata. iCommands are installed automatically on a server install. Documentation is available at docs.irods.org.

baton
Developed by the Wellcome Trust Sanger Institute, baton is a set of command line tools and an API made to ease programmatic access to iRODS. baton provides detailed file and collection management with JSON-formatted listings, as well as a simplified metadata query API, also with JSON-formatted responses. baton is open source software, available on GitHub.

APIs

Java
Jargon is the Java API for iRODS, which provides extensive access to iRODS facilities. *From the DICE Group.*

REST
Provides http-based access to iRODS data and metadata. It is based on Jargon. *From the DICE Group.*

R Client
Based on the REST API, provides access to iRODS data and metadata. *From the DICE Group.*

Python
Provides access to data, user permission, storage resource management, and metadata queries through Python. *From the iRODS Consortium*.

Qt
QRODS is a Qt API for iRODS. It exposes file transfers to/from iRODS, collection management, and metadata management in iRODS. *From the MoDCS Research Group at Universidade Federal de Pernambuco, Brazil.*

C++
The iRODS C++ API is built into iRODS. *From the iRODS Consortium.*

Are there more? Probably.

Keep an eye on iRODS Hub for fresh interfaces as they become available. iRODS Hub is the place to link to clients, rules, dockerfiles, and other software developed for iRODS.

The iRODS Hub is available at: www.irods.org/hub/

Extending iRODS: Plugins

The iRODS architecture is built to be extended with plugins. Plugins are software modules developed independently from one another. They can be installed and configured during runtime.

Plugins make it easy to tailor iRODS to an organization's infrastructure, even as that infrastructure evolves. Each iRODS plugin interface has a variety of compatible plugins, and the iRODS developer community is constantly adding available plugins. There are currently seven pluggable iRODS interfaces.

+ Composable Resources	- Coordinating	- random - roundrobin - compound - replication - load balanced - deferred - passthru
	- Storage	- unixfilesystem - Ceph - WOS - S3 - HPSS - nonblocking - mock archive - universal mass storage
+ Authentication	- native - LDAP (via PAM) - OSAuth - GSI - Kerberos	
+ Network	- TCP - SSL	
+ Database	- PostgreSQL - MySQL - Oracle	
+ API *(Application Programming Interface)*	- various, when needed	
+ Rule Engine	- iRODS Rule Language - Python - Javascript - C++ - Audit (C++)	
+ Microservices	- Nearly 300 commonly used functions	
+ Pluggable Transport *Expected in iRODS 4.3*		

The iRODS Consortium

The iRODS Consortium is a membership foundation that helps organizations get the most out of iRODS.

The Consortium fields a team of software developers, application engineers, and support staff housed at RENCI at the University of North Carolina at Chapel Hill. The iRODS Consortium is responsible for the technology leadership, community development, and support of iRODS software.

Technology Leadership

- The Consortium provides a commercial-grade distribution of iRODS at https://irods.org/download
- Official iRODS documentation is hosted and linked at https://irods.org/documentation
- The iRODS technology roadmap is available at https://irods.org/roadmap

Community Development

- Each year, the Consortium hosts the iRODS User Group Meeting, a symposium that draws 100+ participants to Chapel Hill to share iRODS technologies and case studies.
- The Consortium also hosts iRODS Hub, a site that connects iRODS users to the growing ecosystem of external iRODS Plugins, clients, and other developments related to iRODS.
- The iRODS Consortium maintains a community mailing list, the iRODS chat Google Group.

Support

- The iRODS Consortium provides technical support to all Consortium members and assists commercial members in bringing iRODS-related products and services to market.
- iRODS support is also available through the iRODS Partner Program. iRODS Partners are third party organizations who have demonstrated experience with iRODS.
- The iRODS Consortium also provides several training workshops each year.

Contact Us

Do you need help with your big, important, complex data? Do you need to move it, secure it, process it, share it, or manage it in some other way that you can't right now? Contact the iRODS Consortium to learn more about how we have used iRODS to help organizations like yours.

info@irods.org | www.irods.org/contact

Stay Connected

twitter.com/irods

github.com/irods

youtube.com/irodsconsortium

iRODS

iRODS Technical Overview 2016 edition.