iRODS Primer 2
Integrated Rule-Oriented Data System
Synthesis Lectures on Information Concepts, Retrieval, and Services

Editor
Gary Marchionini, University of North Carolina, Chapel Hill

Synthesis Lectures on Information Concepts, Retrieval, and Services publishes short books on topics pertaining to information science and applications of technology to information discovery, production, distribution, and management. Potential topics include: data models, indexing theory and algorithms, classification, information architecture, information economics, privacy and identity, scholarly communication, bibliometrics and webometrics, personal information management, human information behavior, digital libraries, archives and preservation, cultural informatics, information retrieval evaluation, data fusion, relevance feedback, recommendation systems, question answering, natural language processing for retrieval, text summarization, multimedia retrieval, multilingual retrieval, and exploratory search.

iRODS Primer 2: Integrated Rule-Oriented Data System
Hao Xu, Terrell Russell, Jason Coposky, Arcot Rajasekar, Reagan Moore, Antoine de Torcy, Michael Wan, Wayne Shroeder, and Sheau-Yen Chen
2017

Fuzzy Information Retrieval
Donald H. Kraft and Erin Colvin
2017

Quantifying Research Integrity
Michael Seadle
2016

Incidental Exposure to Online News
Borchuluun Yadamsuren and Sanda Erdelez
2016

Web Indicators for Research Evaluation: A Practical Guide
Michael Thelwall
2016
Trustworthy Policies for Distributed Repositories
Reagan W. Moore, Hao Xu, Mike Conway, Arcot Rajasekar, Jon Crabtree, and Helen Tibbo
2016

The Notion of Relevance in Information Science: Everybody knows what relevance is. But, what is it really?
Tefko Saracevic
2016

Dynamic Information Retrieval Modeling
Grace Hui Yang, Marc Sloan, and Jun Wang
2016

Learning from Multiple Social Networks
Liqiang Nie, Xuemeng Song, and Tat-Seng Chua
2016

Scholarly Collaboration on the Academic Social Web
Daqing He and Wei Jeng
2016

Scalability Challenges in Web Search Engines
B. Barla Cambazoglu and Ricardo Baeza-Yates
2015

Social Informatics Evolving
Pnina Fichman, Madelyn R. Sanfilippo, and Howard Rosenbaum
2015

Mark S. Manasse
2015

Building a Better World with Our Information: The Future of Personal Information Management, Part 3
William Jones
2015

Click Models for Web Search
Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke
2015

Information Communication
Feicheng Ma
2015
<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Media and Library Services</td>
<td>Lorri Mon</td>
<td>2015</td>
</tr>
<tr>
<td>Analysis and Visualization of Citation Networks</td>
<td>Dangzhi Zhao and Andreas Strotmann</td>
<td>2015</td>
</tr>
<tr>
<td>Measuring User Engagement</td>
<td>Mounia Lalmas, Heather O'Brien, and Elad Yom-Tov</td>
<td>2014</td>
</tr>
<tr>
<td>Information and Human Values</td>
<td>Kenneth R. Fleischmann</td>
<td>2013</td>
</tr>
<tr>
<td>Multiculturalism and Information and Communication Technology</td>
<td>Pnina Fichman and Madelyn R. Sanfilippo</td>
<td>2013</td>
</tr>
<tr>
<td>Digital Library Technologies: Complex Objects, Annotation, Ontologies, Classification, Extraction, and Security</td>
<td>Edward A. Fox and Ricardo da Silva Torres</td>
<td>2014</td>
</tr>
<tr>
<td>Children's Internet Search: Using Roles to Understand Children's Search Behavior</td>
<td>Elizabeth Foss and Allison Druin</td>
<td>2014</td>
</tr>
<tr>
<td>Multiculturalism and Information and Communication Technology</td>
<td>Pnina Fichman and Madelyn R. Sanfilippo</td>
<td>2013</td>
</tr>
</tbody>
</table>
Transforming Technologies to Manage Our Information: The Future of Personal Information Management, Part II
William Jones
2013

Designing for Digital Reading
Jennifer Pearson, George Buchanan, and Harold Thimbleby
2013

Information Retrieval Models: Foundations and Relationships
Thomas Roelleke
2013

Key Issues Regarding Digital Libraries: Evaluation and Integration
Rao Shen, Marcos André Goncalves, and Edward A. Fox
2013

Visual Information Retrieval using Java and LIRE
Mathias Lux and Oge Marques
2013

On the Efficient Determination of Most Near Neighbors: Horseshoes, Hand Grenades, Web Search and Other Situations When Close is Close Enough
Mark S. Manasse
2012

The Answer Machine
Susan E. Feldman
2012

Theoretical Foundations for Digital Libraries: The 5S (Societies, Scenarios, Spaces, Structures, Streams) Approach
Edward A. Fox, Marcos André Gonçalves, and Rao Shen
2012

The Future of Personal Information Management, Part I: Our Information, Always and Forever
William Jones
2012

Search User Interface Design
Max L. Wilson
2011

Information Retrieval Evaluation
Donna Harman
2011
<table>
<thead>
<tr>
<th>Title</th>
<th>Authors</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Management (KM) Processes in Organizations: Theoretical Foundations and Practice</td>
<td>Claire R. McInerney and Michael E. D. Koenig</td>
<td>2011</td>
</tr>
<tr>
<td>Search-based Applications: At the Confluence of Search and Database Technologies</td>
<td>Gregory Grefenstette and Laura Wilber</td>
<td>2010</td>
</tr>
<tr>
<td>Information Concepts: From Books to Cyberspace Identities</td>
<td>Gary Marchionini</td>
<td>2010</td>
</tr>
<tr>
<td>Estimating the Query Difficulty for Information Retrieval</td>
<td>David Carmel and Elad Yom-Tov</td>
<td>2010</td>
</tr>
<tr>
<td>Multimedia Information Retrieval</td>
<td>Stefan Rüger</td>
<td>2009</td>
</tr>
<tr>
<td>Online Multiplayer Games</td>
<td>William Sims Bainbridge</td>
<td>2009</td>
</tr>
<tr>
<td>Information Architecture: The Design and Integration of Information Spaces</td>
<td>Wei Ding and Xia Lin</td>
<td>2009</td>
</tr>
<tr>
<td>Reading and Writing the Electronic Book</td>
<td>Catherine C. Marshall</td>
<td>2009</td>
</tr>
<tr>
<td>Hypermedia Genes: An Evolutionary Perspective on Concepts, Models, and Architectures</td>
<td>Nuno M. Guimarãs and Luís M. Carrico</td>
<td>2009</td>
</tr>
</tbody>
</table>
Understanding User-web Interactions via Web Analytics
Bernard J. (Jim) Jansen
2009

XML Retrieval
Mounia Lalmas
2009

Faceted Search
Daniel Tunkelang
2009

Introduction to Webometrics: Quantitative Web Research for the Social Sciences
Michael Thelwall
2009

Exploratory Search: Beyond the Query-response Paradigm
Ryen W. White and Resa A. Roth
2009

New Concepts in Digital Reference
R. David Lankes
2009

Automated Metadata in Multimedia Information Systems: Creation, Refinement, Use in
Surrogates, and Evaluation
Michael G. Christel
2009
iRODS Primer 2
Integrated Rule-Oriented Data System

Hao Xu, Terrell Russell, Jason Coposky, Arcot Rajasekar, Reagan Moore, and Antoine de Torcy
University of North Carolina at Chapel Hill

Michael Wan, Wayne Shroeder, and Sheau-Yen Chen
University of California, San Diego

SYNTHESIS LECTURES ON INFORMATION CONCEPTS, RETRIEVAL, AND SERVICES #57
ABSTRACT

Policy-based data management enables the creation of community-specific collections. Every collection is created for a purpose. The purpose defines the set of properties that will be associated with the collection. The properties are enforced by management policies that control the execution of procedures that are applied whenever data are ingested or accessed. The procedures generate state information that defines the outcome of enforcing the management policy. The state information can be queried to validate assessment criteria and verify that the required collection properties have been conserved. The integrated Rule-Oriented Data System implements the data management framework required to support policy-based data management. Policies are turned into computer actionable Rules. Procedures are composed from a microservice-oriented architecture. The result is a highly extensible and tunable system that can enforce management policies, automate administrative tasks, and periodically validate assessment criteria. iRODS 4.0+ represents a major effort to analyze, harden, and package iRODS for sustainability, modularization, security, and testability. This has led to a fairly significant refactorization of much of the underlying codebase. iRODS has been modularized whereby existing iRODS 3.x functionality has been replaced and provided by small, interoperable plugins. The core is designed to be as immutable as possible and serve as a bus for handling the internal logic of the business of iRODS. Seven major interfaces have been exposed by the core and allow extensibility and separation of functionality into plugins.

KEYWORDS
data life cycle, data grid, digital library, preservation environment, policy-based data management, rule engine, iRODS, metadata catalog, assessment criteria, policies, microservices
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>xv</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Integrated Rule-Oriented Data System</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Data Grid Overview</td>
<td>4</td>
</tr>
<tr>
<td>3 iRODS Architecture</td>
<td>13</td>
</tr>
<tr>
<td>3.1 Virtualization in iRODS</td>
<td>17</td>
</tr>
<tr>
<td>3.2 iRODS Components</td>
<td>18</td>
</tr>
<tr>
<td>3.3 User Environment Variables</td>
<td>19</td>
</tr>
<tr>
<td>3.4 Configuration Files</td>
<td>19</td>
</tr>
<tr>
<td>3.4.1 ~/.odbc.ini</td>
<td>19</td>
</tr>
<tr>
<td>3.4.2 ~/.irods/.irodsA</td>
<td>19</td>
</tr>
<tr>
<td>3.4.3 /etc/irods/server_config.json</td>
<td>20</td>
</tr>
<tr>
<td>3.4.4 ~/.irods/irods_environment.json</td>
<td>23</td>
</tr>
<tr>
<td>3.4.5 Checksum Configuration</td>
<td>25</td>
</tr>
<tr>
<td>3.4.6 Special Characters</td>
<td>26</td>
</tr>
<tr>
<td>3.5 Plugin Interfaces</td>
<td>26</td>
</tr>
<tr>
<td>3.5.1 Pluggable Microservices</td>
<td>27</td>
</tr>
<tr>
<td>3.5.2 Composable Resources</td>
<td>27</td>
</tr>
<tr>
<td>3.5.3 Pluggable Authentication</td>
<td>28</td>
</tr>
<tr>
<td>3.5.4 Pluggable Network</td>
<td>28</td>
</tr>
<tr>
<td>3.5.5 Pluggable Database</td>
<td>29</td>
</tr>
<tr>
<td>3.5.6 Pluggable RPC API</td>
<td>29</td>
</tr>
<tr>
<td>3.5.7 Pluggable Rule Engine</td>
<td>30</td>
</tr>
<tr>
<td>3.6 Example Plugins</td>
<td>30</td>
</tr>
<tr>
<td>3.6.1 Composable Resources</td>
<td>30</td>
</tr>
<tr>
<td>3.6.2 Pluggable Authentication</td>
<td>40</td>
</tr>
<tr>
<td>4 Rule-Oriented Programming</td>
<td>51</td>
</tr>
<tr>
<td>4.1 Session State Variables</td>
<td>55</td>
</tr>
<tr>
<td>4.2 Persistent State Information Variables</td>
<td>55</td>
</tr>
</tbody>
</table>
6.3 Examples ................................................................. 103
  6.3.1 The Plugin Factory .............................................. 104
  6.3.2 The Microservice Definition ................................. 105
  6.3.3 Building and Installing the Example Code ............... 107
  6.3.4 Testing the Microservice ...................................... 107

A Exercises ................................................................. 109
  A.1 Short Questions .................................................. 109
  A.2 Essay Questions .................................................. 111

Authors’ Biographies .................................................... 113
Acknowledgments

This research was supported by:

- NSF SDCI 0721400, SDCI Data Improvement: Data Grids for Community Driven Applications (2007–2010)

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Archives and Records Administration, the National Science Foundation, or the U.S. Government.

Hao Xu, Terrell Russell, Jason Coposky, Arcot Rajasekar, Reagan Moore, Antoine de Torcy, Michael Wan, Wayne Shroeder, and Sheau-Yen Chen
March 2017
CHAPTER 1

Introduction

Recent decades have seen a rapid rise in collaborative activities in scientific research, and more broadly across many sectors of society. Driven by new information technologies such as the Web as well as the increasing complexity and interdisciplinary nature of today's research problems, from climate change to the world's increasingly integrated economies, the need for technologies, sometimes called "cyberinfrastructure," that enable researchers to collaborate effectively continues to grow rapidly. The Integrated Rule-Oriented Data System (iRODS) is a state-of-the-art software that supports collaborative research, and, more broadly, management, sharing, publication, and long-term preservation of data that are distributed.

A tool for collaboration, iRODS is itself the product of a fruitful collaboration spanning more than two decades among high-performance computing (HPC), preservation, and library communities, whose real-world needs have driven and shaped iRODS development. The computational science and HPC communities are inherently interdisciplinary, generating and using very large data collections distributed across multiple sites and groups. The massive size of these data collections has encouraged development of unique capabilities in iRODS that allow scaling to collections containing petabytes of data and hundreds of millions of files.

The preservation community brings the need for long-term preservation of digital information, a challenging problem that is still an active research area to which iRODS research activities have made significant contributions. Interestingly, there turned out to be significant commonalities in the requirements for preserving digital data in time and collaborative sharing of distributed data collections across space, whether geographic, institutional, disciplinary, etc.

The third community that has contributed to iRODS development is the library community, with expertise in descriptive metadata that is essential for management, discovery, repurposing, as well as controlled sharing and long-term preservation of digital collections.

In collaborating with these communities, iRODS research and development has been characterized by close attention to the practical requirements of a wide range of users, resulting in pioneering architecture and solutions to numerous distributed data challenges that now form the iRODS Data System for managing, sharing, publishing, and preserving today's rapidly growing and increasingly complex digital collections.

iRODS is a software middleware, or "cyberinfrastructure," that organizes distributed data into a sharable collection. The iRODS software is used to implement a data grid that assembles data into a logical collection. Properties such as integrity (uncorrupted record), authenticity (linking of provenance information to each record), chain of custody (tracking of location and
2 1. INTRODUCTION

management controls within the preservation environment), and trustworthiness (sustainability of the records) can be imposed on the logical collection. When data sets are distributed across multiple types of storage systems, across multiple administrative domains, across multiple institutions, and across multiple countries, data grid technology is used to enforce uniform management policies on the assembled collection. The specific challenges addressed by the iRODS Data Grid include the following:

• Management of interactions with storage resources that use different access protocols. The data grid provides mechanisms to map from the actions requested by a client to the protocol required by a specific vendor supplied disk, tape, archive, or object-relational database.

• Support for authentication and authorization across systems that use different identity management systems. The data grid authenticates all access, and authorizes and logs all operations on the files registered into the shared collection.

• Support for uniform management policies across institutions that may have differing access requirements such as different institutional research board approval processes. The policies controlling use, distribution, replication, retention, disposition, authenticity, integrity, and trustworthiness are enforced by the data grid.

• Support for wide-area-network access. To maintain interactive response, network transport is optimized for moving massive files (through parallel input/output (I/O) streams), for moving small files (through encapsulation of the file in the initial data transfer request), for moving large numbers of small files (aggregation into tar files), and for minimizing the amount of data sent over the network (execution of remote procedures such as data subsetting on each storage resource).

In response to these challenges, iRODS is an ongoing research and software development effort to provide software infrastructure solutions that enable collaborative research. The software systems are implemented as middleware that interacts with remote storage systems on behalf of the users. The goal of the iRODS Consortium is to develop generic software that can be used to implement all distributed data management applications, through changing the management policies and procedures. This has been realized by creating a highly extensible software infrastructure that can be modified without requiring the modification of the core software or development of new software code.

This publication describes the data grid technology in Chapter 2, the iRODS architecture in Chapter 3, the Rule-Oriented Programming model in Chapter 4, the iRODS Rule system in Chapter 5, and the iRODS microservices in Chapter 6.

Documentation for iRODS is continually being updated by the growing iRODS open source community on the iRODS website at http://www.irods.org/, covering topics such as installation, how to use iRODS, administration, and development information. The website also contains iRODS-related publications for further reading.
CHAPTER 2

Integrated Rule-Oriented Data System

iRODS is software middleware that manages a highly controlled collection of distributed digital objects, while enforcing user-defined Management Policies across multiple storage locations. The iRODS system is generic software infrastructure that can be tuned to implement any desired data management application, ranging from a Data Grid for sharing data across collaborations, to a digital library for publishing data, to a preservation environment for long-term data retention, to a data processing pipeline, to a system for federating real-time sensor data streams.

The iRODS technology is originally developed by Data Intensive Cyber Environments (DICE) Center at University of North Carolina at Chapel Hill and University of California, San Diego, and is currently developed by the iRODS Consortium, housed within the Renaissance Computing Institute (RENCI) at the University of North Carolina at Chapel Hill.

iRODS 4.0+ represents a major effort to analyze, harden, and package iRODS for sustainability, modularization, security, and testability. This has led to a fairly significant refactorization of much of the underlying codebase. iRODS has been modularized whereby existing iRODS 3.x functionality has been replaced and provided by small, interoperable plugins. The core is designed to be as immutable as possible and serve as a bus for handling the internal logic of the business of iRODS (data storage, policy enforcement, etc.). Seven major interfaces have been exposed by the core and allow extensibility and separation of functionality into plugins. A few plugins are included by default in iRODS to provide a set of base functionality.

The ideas for the iRODS project have existed for a number of years, and became more concrete through the National Science Foundation-funded project, Constraint-Based Knowledge Systems for Grids, Digital Libraries, and Persistent Archives, which began in the Fall of 2004. The development of iRODS was driven by the lessons learned in nearly 10 years of deployment and production use of the DICE Storage Resource Broker (SRB) Data Grid technology and through applications of theories and concepts from a wide range of well-known paradigms from computer science fields such as active databases, program verification, transactional systems, logic programming, business rule systems, constraint-management systems, workflows, and service-oriented architecture. The iRODS Data Grid is an adaptable middleware, in which management policies and management procedures can be dynamically changed without having to rewrite software code.
2. INTEGRATED RULE-ORIENTED DATA SYSTEM

The iRODS Data Grid expresses management policies as computer actionable rules, and management procedures as sets of remotely executable microservices. The rules control the execution of the microservices. The state information generated by the microservices is stored in a metadata catalog (iCAT). The iRODS Data Grid manages input and output information from the microservices (81 Session Variable Attributes and 109 Persistent State Information Attributes), manages composition of 200+ microservices into Actions that implement the desired management procedures, and enforces 80+ active rules while managing a Distributed Collection. An additional set of eight alternate rules is provided as examples of the tuning of Management Policies to specific institutional requirements. The rules and microservices are targeted toward data management functions needed for a wide variety of data management applications. The open source iRODS Data Grid is highly extensible, supporting dynamic updates to the Rule Base, the incorporation of new microservices, the addition of new Persistent State Information, as well as the use of additional or updated plugins. With the knowledge provided by this document, a reader will be able to add new rules, create new microservices, and build a data management environment that enforces their institutional Management Policies and procedures.

2.1 DATA GRID OVERVIEW

The iRODS technology builds upon the lessons learned from the first generation of data grid technology developed by the DICE group, the SRB. The same basic concepts needed for distributed data management and organization of distributed data into sharable collections that were implemented in the SRB have also been implemented in iRODS.

The DICE SRB Data Grid is a software infrastructure for sharing data and metadata distributed across heterogeneous resources using uniform Application Programming Interfaces (APIs) and Graphical User Interfaces. To provide this functionality, the SRB abstracts key concepts in data management: data object names, and sets of data objects, resources, users, and groups, and provides uniform methods for interacting with the concepts. The SRB hides the underlying physical infrastructure from users by providing global, logical mappings from the digital entities registered into the shared collection to their physical storage locations. Hence, the peculiarities of storage systems and their access methods, the geographic or administrative location of data, and user authentication and authorization across systems, are all hidden from the users. A user can access files from an online file system, near-line tapes, relational databases, sensor data streams, and the Web without having to worry about where they are located, what protocol to use to connect and access the system, and without establishing a separate account or password/certificate to each of the underlying computer systems to gain access. These virtualization mechanisms are implemented in the SRB system by maintaining mappings and profile metadata in a permanent database system called the MCAT Metadata Catalog and by providing integrated data and metadata management, which links the multiple subsystems in a seamless manner.

A key concept is the use of Logical Name Spaces to provide uniform names to entities located in different administrative domains and possibly stored on different types of storage resources.
When we use the term Logical Name Space, we mean a set of names that are used by the Data Grid to describe entities. Logical Name Spaces are used to describe the users (user Logical Name Space), the files (file Logical Name Space), and storage resources (resource Logical Name Space). An implication is that the Data Grid must maintain a mapping from the logical names to the names understood by each of the remote storage locations. All operations within the iRODS Data Grid are based on the iRODS Logical Name Spaces. The iRODS system internally performs the mapping to the physical names, and issues operations on behalf of the user at the remote storage location.

Note that the original SRB Data Grid defined three Logical Name Spaces.

1. **Logical names for users.** Each person is known to the Data Grid by a unique name. Each access to the system is authenticated based on either a public key certificate, Kerberos certificates, shared secret, or other token-based authentication (via PAM).

2. **Logical names for files and collections.** The Data Grid supports the logical organization of the distributed files into a hierarchy that can be browsed. A logical collection can be assembled in which files are logically grouped together even though they reside at different locations.

3. **Logical names for storage resources.** The Data Grid can organize resources into hierarchies, and apply operations on those hierarchies. An example is the random resource, in which files are distributed randomly across multiple storage systems. An even more interesting example is the dynamic addition of a new storage resource to a hierarchy, replication of live data to that new storage resource, and then removal of a legacy storage system transparently to the users of the system.

Both the SRB and iRODS Data Grids implement Logical Name Spaces for users, files, and storage resources. The best example to start with is the logical names for files and directories in iRODS: the Data Object and Collection names. Each individual file stored in iRODS has both a logical and physical path and name. The logical names are the collection and dataObject names as they appear in iRODS. These are the names that users can define and see when accessing the iRODS Data Grid.

The iRODS system keeps track of the mapping of these logical names to the physical files (via storage of the mapping in the iCAT Metadata Catalog). Within a single collection, the individual data objects might exist physically on separate file systems and even on separate physical servers. The iRODS system automatically updates the mappings whenever operations are performed on the files, and enables users to access the files (if they have the appropriate authorization) regardless of where the files are physically located.

This is a form of “infrastructure independence,” which is essential for managing distributed data. The user or administrator can move the files from one storage file system (Resource) to another, while the logical name the users see remains the same. An old storage system can be replaced by a new one with the physical files migrated to the new storage system. The iRODS...
system automatically tracks the changes for the users, who continue to reference the files by the persistent and user-definable Logical Name Space.

The following example illustrates this with the iRODS iCommands (Unix-style shell commands that are executed from a command line prompt). Explanatory comments are added after each shell command as a string in parentheses. The command line prompt is “host$” in this example. The commands are shown in *italics*. The output is shown in normal.

```
host$ imkdir t1 (Make a new subcollection t1)
host$ icd t1 (Make t1 the current default working directory)
host$ iput file1 (Store a file into iRODS into the working directory)
host$ ils (Show the files in iRODS, that is the logical file names)

/zz/home/rods/t1:
  file1

host$ ils -l (Show more detail, including the logical resource name)

/zz/home/rods/t1:
  rods 0 demoResc 18351 2017-01-17.12:22 & file1

host$ ils -L (Show more detail, including the physical path where the file was stored)

/zz/home/rods/t1:
  rods 0 demoResc 18351 2017-01-17.12:22 & file1
  /scratch/slocal/rods/iRODS/Vault/home/rods/t1/file1

  The first item on the *ils* output line is the name of the owner of the file (in this case, “rods”). The second item is the replication number, which we further explain below. The third item is the Logical Resource Name. The fourth item is the size of the file in bytes. The fifth item is the date. The sixth item (“&”) indicates the file is up-to-date. If a replica is modified, the “&” flag is removed from the out-of-date copies.

  In the example above, the iRODS logical name for the file was “file1” and the file was stored in the logical collection “/zz/home/rods/t1”. The original physical file name was also “file1”. The logical resource name was “demoResc”. When iRODS stored a copy of the file onto the storage resource “demoResc”, the copy was made at the location:

  /scratch/slocal/rods/iRODS/Vault/home/rods/t1/file1

  Any storage location at which an iRODS Server has been installed can be used for the repository through the “-R” command line option. Even though the example below stores “file2”
on storage resource “demoRescQe2”, both “file1” and “file2” are logically organized into the same logical collection “/zz/home/rods/t1”.

```bash
host$ iput -R demoRescQe2 file2 (Store a file on the `demoRescQe2` vault/host)
host$ ils

/zz/home/rods/t1:
  file1
  file2

host$ ils -l

/zz/home/rods/t1:
  rods 0 demoResc 18351 2017-01-17.12:22 & file1
  rods 0 demoRescQe2 64316 2017-01-17.12:29 & file2

host$ ils -L

/zz/home/rods/t1:
  rods 0 demoResc 18351 2017-01-17.12:22 & file1
  /scratch/slocal/rods/iRODS/Vault/home/rods/t1/file1
  rods 0 demoRescQe2 64316 2017-01-17.12:29 & file2
  /scratch/qe2/iRODS/Vault/home/rods/t1/file2
```

Other operations can be performed on files.

- **Registration** is the creation of iRODS metadata that point to the file without making a copy of the file. The `ireg` command is used instead of `iput` to register a file. In the example below, “file3a” is added to the logical collection. Note that its physical location remains the original file system (“/users/u4/test/file3”) and a copy was not made into the iRODS Data Grid.

```bash
host$ ireg /users/u4/test/file3 /zz/home/rods/t1/file3a
host$ ils

/zz/home/rods/t1:
  file1
  file2
  file3a

host$ ils -l
```
• Replication is the creation of multiple copies of a file on different physical resources. Note that the replication is done on a file that is already registered or put into an iRODS logical collection.

host$ irepl -R demoRescEe2 file1
host$ ils

/zz/home/rods/t1:
    file1
    file1
    file2
    file3a

host$ ils -l

/zz/home/rods/t1:
    rods 0 demoResc 18351 2017-01-17.12:22 & file1
    rods 1 demoRescEe2 18351 2017-01-17.12:33 & file1
    rods 0 demoRescEe2 64316 2017-01-17.12:29 & file2
    rods 0 demoResc 10627 2017-01-17.12:31 & file3a