

iRODS Client: Zone Management Tool (ZMT)

Bo Zhou

Renaissance Computing
Institute (RENCI)
UNC Chapel Hill
zbo@renci.org

Jason Coposky

Renaissance Computing
Institute (RENCI)
UNC Chapel Hill
jasonc@renci.org

Terrell Russell

Renaissance Computing
Institute (RENCI)
UNC Chapel Hill
unc@terrellrussell.com

ABSTRACT

The iRODS Zone Management Tool (ZMT) is a new client that uses the iRODS C++ REST API. It has a design goal of handling the administrative side of running an iRODS Zone (managing users/groups/resources, etc.). This paper will introduce the ZMT, current status, and future work.

Keywords

iRODS, data management, administration, REST, reactjs

INTRODUCTION

The iRODS ecosystem has historically been driven by user demand and much community development. Over the years, this has led to a number of different software clients issuing requests to the iRODS server. Some of those clients have been primarily for **rodsuser**-level operations, and others may have included some **rodsadmin**-level operations.

This new client will be **rodsadmin**-only, and is designed to provide a baseline and future home for a single-pane-of-glass to administer an entire iRODS Zone. ZMT aims to include managing users and groups and resources, but also server configuration itself, including policy editing, deployment, and rollback.

This paper provides a first look at the work so far.

DESIGN GOALS

The iRODS community needs a user-friendly, easy to deploy, fast, and light-weight web-based graphical user interface (GUI) to manage an iRODS Zone. The iRODS Zone Management Tool[1] meets those design goals.

ZMT is based on the React.JS[2] single page application (SPA) framework and speaks only HTTP/HTTPS to the configured iRODS C++ REST API[3] endpoint. It exposes existing configuration of all iRODS Servers in a Zone and an administration endpoint to manipulate system elements similar to **iadmin**.

It is envisioned that ZMT will assume the administration duties of Metalnx[4] which should free that codebase of some of its historical complexity. As a **rodsadmin**-only tool, ZMT will continue to focus on administrator concerns.

CONFIGURATION

ZMT configuration is handled by a single environment file that is loaded at application startup. This **.env** file provides read-only variables that can control different aspects of a ZMT deployment.

The minimum required configuration involves a single value representing the iRODS C++ REST API endpoint for the iRODS Zone that ZMT is being used to manage:

```
REACT_APP_REST_API_URL=protocol://host:port
```

Additional configuration options are available to control which port ZMT itself runs on and how the application looks. A `sample.env` file is included in the repository for reference:

```
LISTEN_PORT=3000
REACT_APP_REST_API_URL=protocol://host:port
REACT_APP_APPBAR_LOGO=iRODS-logo.jpg
REACT_APP_LOGIN_LOGO=iRODS-logo-1.png
REACT_APP_BRANDING_NAME=Zone Management Tool
REACT_APP_PRIMARY_COLOR=#04bdaf
REACT_APP_SECONDARY_COLOR=#ffffff
```

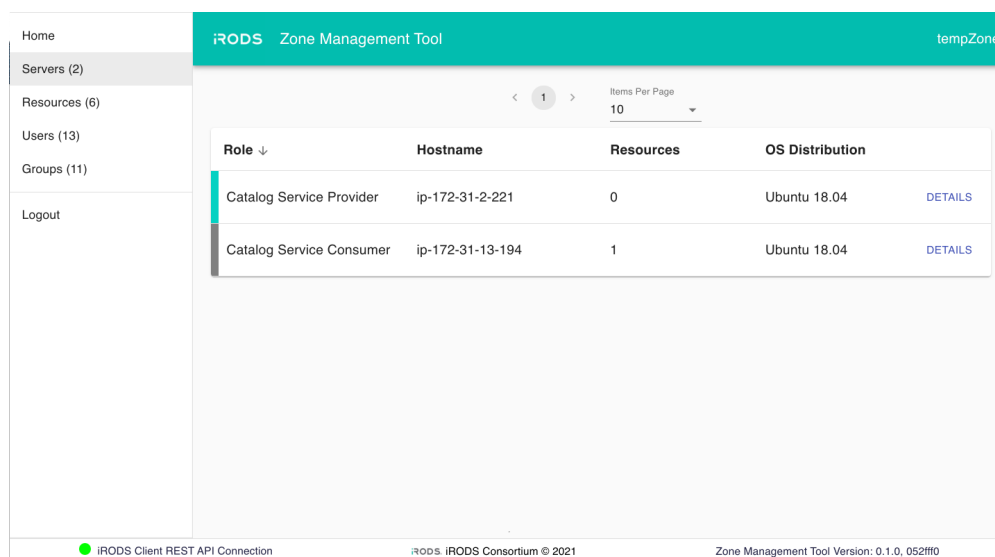
ENDPOINTS

The initial release of the ZMT includes four endpoints, represented in the left sidebar as sections of the web application. Each section provides visibility over and affords management of a particular 'noun' in the iRODS Zone's namespace. These include `/servers`, `/resources`, `/users`, and `/groups`. The `/` (or `/home`) endpoint serves as an overview or dashboard and will be populated in future work.

`/servers`

The `/servers` section provides a table view (Figure 1) of basic server information in the local iRODS Zone. Displayed columns include, for each server, the role (Catalog Service Provider or Catalog Service Consumer), the hostname, the number of attached storage resources, and the operating system and version of that machine.

The data in this table is gathered from the results of querying the iRODS REST C++ REST `/zone_report` endpoint (similar to the output from `izonereport`). The table provides paging for a large number of servers and each column is sortable in two directions.



Role ↓	Hostname	Resources	OS Distribution	
Catalog Service Provider	ip-172-31-2-221	0	Ubuntu 18.04	DETAILS
Catalog Service Consumer	ip-172-31-13-194	1	Ubuntu 18.04	DETAILS

Figure 1. ZMT `/servers` - Table View

More details about each server are available in the Details View (Figure 2). This information includes the contents of each server's `server_config.json` file. Future work will allow for manipulation of the server information, but it is not editable via ZMT at this time.

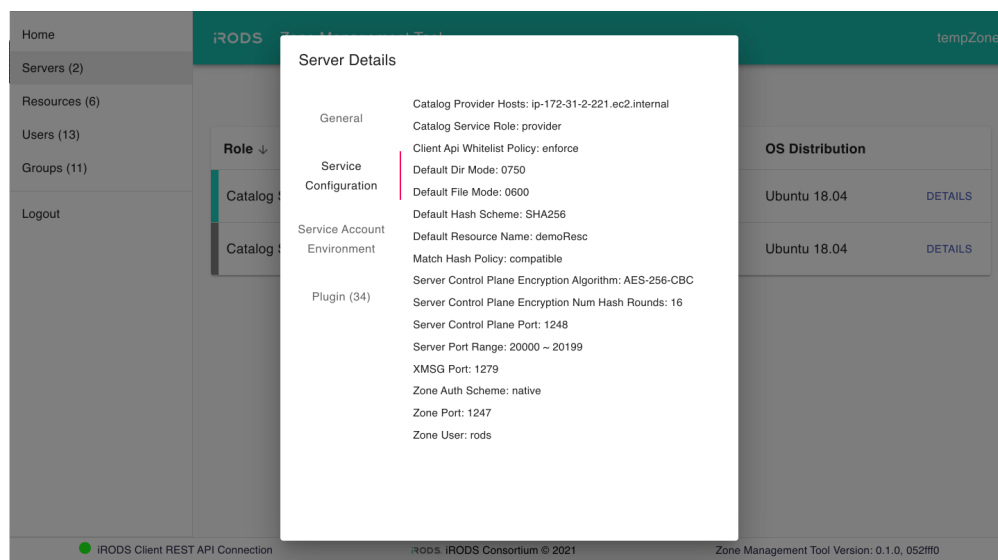


Figure 2. ZMT /servers - Details View

/resources

The `/resources` section provides two different views.

Similar to `/servers`, the table view (Figure 3) is a sortable set of columns including the name, type, associated hostname, and vault path of each resource. The listing can be live filtered by name and provides functionality for creating, renaming, removing, and otherwise updating resources (including context strings).

Name ↑	Type ↑	Hostname ↑	Vault Path ↑
Ufss3	unixfilesystem	ip-172-31-5-160	/tmp/test/Ufss3
compound1	compound	ip-172-31-5-160	/tmp/test/
compound2	compound	ip-172-31-5-160	/tmp/test/
deMoRESCmoremore	unixfilesystem	EMPTY_RESC_HOST	/tmp/test/
demoResc	unixfilesystem	EMPTY_RESC_HOST	/var/lib/irods/Vault
mockarchive1	mockarchive	ip-172-31-5-160	/tmp/test/
ufs1	unixfilesystem	localhost	/tmp/ufs1vault
ufs3	unixfilesystem	ip-172-31-5-160	/tmp/test/uFS2

Figure 3. ZMT /resources - Table View

The tree view (Figure 4) shows the resource hierarchies of the connected Zone with the parent-child relationships between the composable resources[5].

The tree provides drag-and-drop editing capability of these parent-child relationships. A series of edits can be staged and reviewed before being undone or saved to the server. This allows an administrator to minimize the window of time when the resource hierarchies are being updated to their new target state.

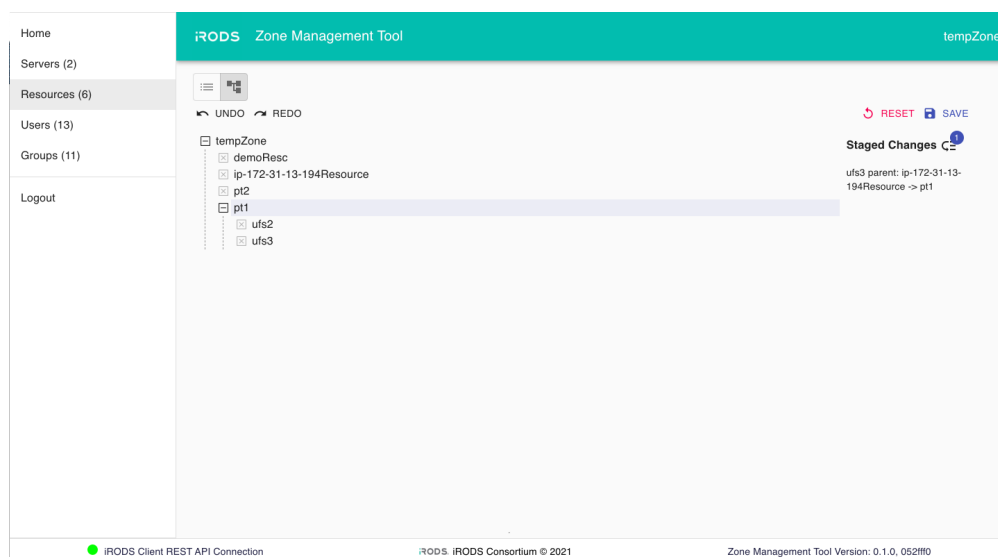


Figure 4. ZMT /resources - Tree View

/users

The `/users` section lists the users defined in the local Zone's catalog (Figure 5). This includes both local and remote (federated) users. The table can be paged and is filterable. New users can be created and users can be removed.

Username ↑	Type ↑	Action
another	rodsadmin	EDIT REMOVE
caSeInSENSITIVE	rodsuser	EDIT REMOVE
hello	rodsuser	EDIT REMOVE
rods	rodsadmin	
rods1	rodsadmin	EDIT REMOVE
rods2	groupadmin	EDIT REMOVE
t1	groupadmin	EDIT REMOVE
test	rodsuser	EDIT REMOVE

Figure 5. ZMT `/users` - Table View

When editing a particular user (Figure 6), the existing groups are listed and whether this user is a member of each group. A user can be added or removed from groups in this view.

Group Name	Status	Action
RENC1	Not in group	ADD
ReNcI	Not in group	ADD
public	In group	REMOVE
renci	Not in group	ADD
rmENC1	In group	REMOVE
testGroup1	Not in group	ADD
testGroup2	Not in group	ADD
testGroup3	Not in group	ADD
testGroup4	Not in group	ADD
testGroup5	Not in group	ADD

Figure 6. ZMT `/users` - Editing Group Membership

/groups

The **/groups** section functions similarly to **/users**. The table view (Figure 7) lists group name and the number of users in each group and allows for editing a group's membership. The table can be paged and is filterable. New groups can be created and groups can be removed.

Group Name ↑	Users	Action
RENCi	2	EDIT REMOVE
ReNci	1	EDIT REMOVE
public	6	EDIT
renci	0	EDIT REMOVE
rrrRENCi	2	EDIT REMOVE
testGroup1	0	EDIT REMOVE
testGroup2	0	EDIT REMOVE
testGroup3	0	EDIT REMOVE
testGroup4	0	EDIT REMOVE

Figure 7. ZMT /groups - Table View

When editing a particular group (Figure 8), the existing users are listed and whether they currently belong to this group. Users can be added or removed from the group in this view.

User Name	Type	Status	Action
hello	rodsuser	Not in RENCi	ADD
test	rodsuser	Not in RENCi	ADD
rods1	rodsadmin	Member of RENCi	REMOVE
t1	groupadmin	Member of RENCi	REMOVE
another	rodsadmin	Not in RENCi	ADD
rods	rodsadmin	Not in RENCi	ADD
caSelnsENSITIVE	rodsuser	Not in RENCi	ADD
rods2	groupadmin	Not in RENCi	ADD

Figure 8. ZMT /groups - Editing Group Membership

iRODS REST API Endpoint Connection

The footer of the ZMT shows the iRODS REST API endpoint connection indicator. The ZMT uses four REST API endpoints to provide the different views (`/admin`, `/auth`, `/query`, and `/zone_report`). The overlay provides status information about whether those REST API endpoints are alive and responding.

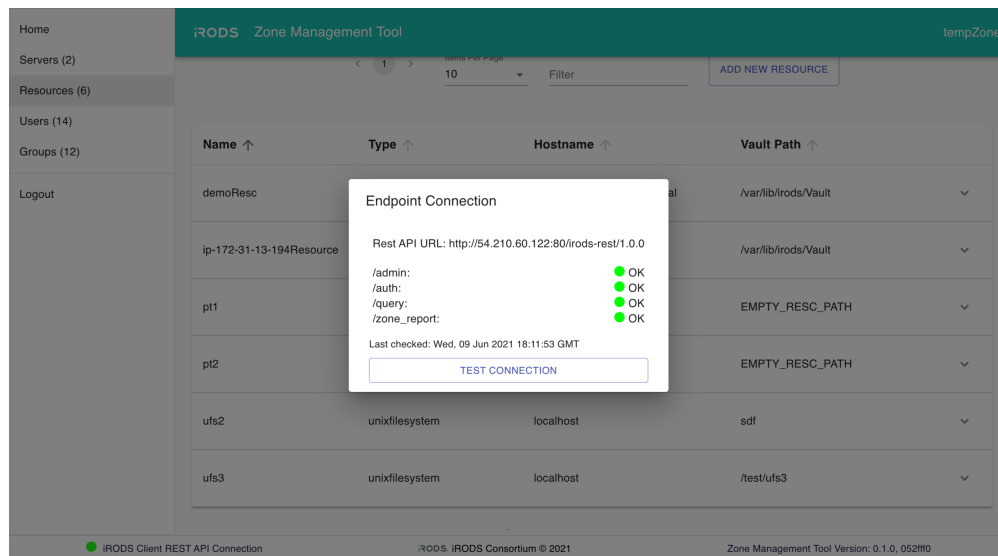


Figure 9. endpoint-connection

FUTURE WORK

As this is a first look at a new React-based iRODS administrative GUI, there are many features on the roadmap. Future capabilities will include management functions for the delay queue, tickets, and remote zones. There will be more monitoring provided for the different servers and their storage. These may include both up/down status indicators as well as health checks for some (un-)common cases that cause trouble for administrators.

In the farther future, we hope that ZMT will grow the ability to manage the server configuration directly, including policy sets and higher order capabilities.

SUMMARY

This paper provides a vision and describes a pre-release of the new iRODS Zone Management Tool (ZMT). Basic administration of resources, users, and groups is covered while connected to the iRODS C++ REST API. As this interface gains additional functionality, it will allow other applications to simplify and remove some of their `rodsadmin` complexity.

REFERENCES

- [1] Zone Management Tool (ZMT). https://github.com/irods/irods_client_zone_management_tool
- [2] React - A JavaScript library for building user interfaces. <https://reactjs.org/>
- [3] Coposky, Jason; Russell, Terrell; iRODS Client: C++ REST API (2021) <https://irods.org/uploads/2021/Coposky-iRODS-C-Plus-Plus-REST-API-paper.pdf>
- [4] Zhou, Bo; Draughn, Kory; Coposky, Jason; Russell, Terrell; Conway, Mike; iRODS Client: Metalnx 2.4.0 with GalleryView (2021) https://irods.org/uploads/2021/Zhou-iRODS-Metalnx-2.4.0_with_GalleryView-slides.pdf
- [5] Russell, Terrell; Coposky, Jason; Johnson, Harry; Idaszak, Ray; Schmitt, Charles; E-iRODS Composable Resources (2013). iRODS User Group Meeting 2013. <https://irods.org/uploads/2013/02/eirods-composable-resources.pdf>