

Administrators often ask how they can archive data in CA-AC (Rally) without losing visibility to it until they no longer need the data for reporting/historical purposes. This purpose of this document is to outline the various methods available.

Archiving Overview

The three main methods of archiving data all revolve around the permissions users have to the [project\(s\)](#) and the need the business has for visibility to that data. Most often teams and businesses need visibility into the data for different periods of time and for different reasons. However, both needs can be easily accommodated, and when the data no longer needs to be visible it can be completely hidden without removing it in any permanent sense

We'll approach the methods as follows

- Team Archival
 - Sprints, Releases, Test Cases, Defects
- Greater Business Archival
 - Portfolios, Programs, Value Streams
- Final Archival

Team Archival

Teams often want to have access to data from the last six months to a year for the purposes of retrospectives, reporting, and historical references to testing, defects, etc. Usually though, the business wants to be able to lock the data down to ensure no human error alters the historical data. How do we accomplish give the team what it wants while also providing the security the Business needs?

How do we manage this within the existing [project hierarchy](#) and not alter the way the data rolls up and reports from the archive?

First we need to understand that the team has a requirement for the data to remain live for a certain time from completion/acceptance. It's the Team that needs to agree on how long they keep the data within the team level; e.g. 3 month, six months, etc.

However, the team cannot dictate a timeframe shorter than the business may need it at the team level.

***most often data is moved out of the team hierarchy when that team is no longer working on that product/program/value stream*

Once the team has decided on what the cutoff date is on the data the Project Admin or Workspace Admin can

1. Rename the old team project to zArchive
 - a. z allows the sort to place it at the bottom of the hierarchy, keeping the list visually easier for the teams
2. Create a new project in the existing hierarchy for all development work going forward
 - a. Team contributors will then copy over any test cases and duplicate defects
 - b. Never move the work item to the new project or else you will lose the reporting data for that object in its historical reference.
3. Once the data that needs to be copied over (not moved) the Project or Workspace admin can then flag all users as Viewers only to the old team project so that they have visibility but not one can edit the data.
 - a. This locks the data down, any requests to then edit the data needs to be routed through the proper admins with proper justifications (business needs that have been approved)

Before

Parent Project

- Team Alpha
- Team Beta

During

Parent Project

- Team Alpha (all users with original permissions)
- Team Beta
- zAlpha Archive (only those users needed to copy and duplicate have editor access - all others changed to Viewers)

After

Parent Project

- Team Alpha (*all users with original permissions*)
- Team Beta
- zAlpha Archive (*all users set to Viewers*)

Business Archival

There comes a time when the teams no longer need to see the data from their past work but the business still needs to see the data. How do we accommodate this need without disrupting the data or kluding up the view of the team's hierarchy?

This requires the Workspace Admin to get involved and create a new branch on the organization for an Archive repository. Now it could be strung under the parent layer or it could be it's own, peer, organization of data.

There are values to each are based on the needs of the business:

1. If the need is to see the archived data within the whole of the development organization then the Archive Node should reside under the Parent level
2. If the data does not need to be seen within the current development then a peer node is best

Parented Archive

Parent Project
- zArchive

Peer Node

Parent Project
- Child
zArchive
- Archived Child

Regardless of which serves your particular needs best the process is still the same, just the scope and scale is a little different as you need to also address the Product layer of the organization and what their needs are to access and view the data over time.

Again, something that should be worked out with the business but there are the following items that need to be considered above the Stories, Defects, and test cases at the team level; Portfolio Items and Milestones have to be assessed and managed as to what belongs in Archive and what is still active and needed in other layers of the organization.

Once the business and product needs are understood the Workspace or the Subscription Admin can then

1. Create a New project called zArchive and place it where appropriate on the hierarchy
2. Set a deadline for access to the data for editors
 - a. Inform users from those affected layers that the data will be moving to the Archive repository in the hierarchy and to prep any data necessary
 - b. Moving PIs and reorganizing the Milestones as/if needed
3. Once the data has been prepared, using the Administrative tools in Setup, change the actual parent of the Archived projects to be moved, to the zArchived node
4. Set all users permissions to Viewer only for only those user who still, need to access the data
 - a. This should not be anyone from the teams at this point in the process (exceptions do occur)

Prior to Archiving (in hierarchy)

Parent Project
- Child 1 (*all users with original permissions*)
- Strategy And Product
- Child 2 (*all users with original permissions*)
- Team Alpha (*all users with original permissions*)
- Team Beta
- zAlpha Archive (*all users set to Viewers*)

During Archiving (in hierarchy)

Parent Project

- Child 1 (*all users with original permissions*)
 - Strategy And Product (*all users with original permissions*)
 - zArchive S&P (*only those who need to prep data have original permissions*)
- Child 2 (*all users with original permissions*)
 - Team Alpha (*all users with original permissions*)
 - Team Beta
 - zAlpha Archive (*all users set to Viewers*)
- zArchive (only users who need editor's rights)

After Archiving (in hierarchy)

Parent Project

- Child 1 (*all users with original permissions*)
 - Strategy And Product (*all users with original permissions*)
- Child 2 (*all users with original permissions*)
 - Team Alpha (*all users with original permissions*)
 - Team Beta
- zArchive (only users who need editor's rights or Viewers rights)
 - zArchive S&P
 - zAlpha Archive

Final Archival

This is the easiest of them all; once it's been determined that there is no longer a need for access to the data the Workspace or Subscription admin can get into the Setup and close those Archived projects that no longer need to be visible. This will remove the projects for view or access until such time as the admin reopens the project.

If you have any questions on this material or would like to have a conversation regarding what process may be best for your needs, please reach out to your [Customer Success Manager](#) for assistance.