Power Automate Desktop Locker Management

Dieter De Cock
Microsoft – Power CAT

Purpose

With this solution you will be able to create and use Lockers which store Values for use within Cloud & Desktop Flows.

This solution can be seen as a light-weight alternative for Azure Key Vault available within Power Platform environments.
This solution is entirely built on Dataverse with standard Power Platform components and security features.

This solution is an open-source project, published on GitHub. Any issues, concerns, inquiries... should not be directed to Microsoft.
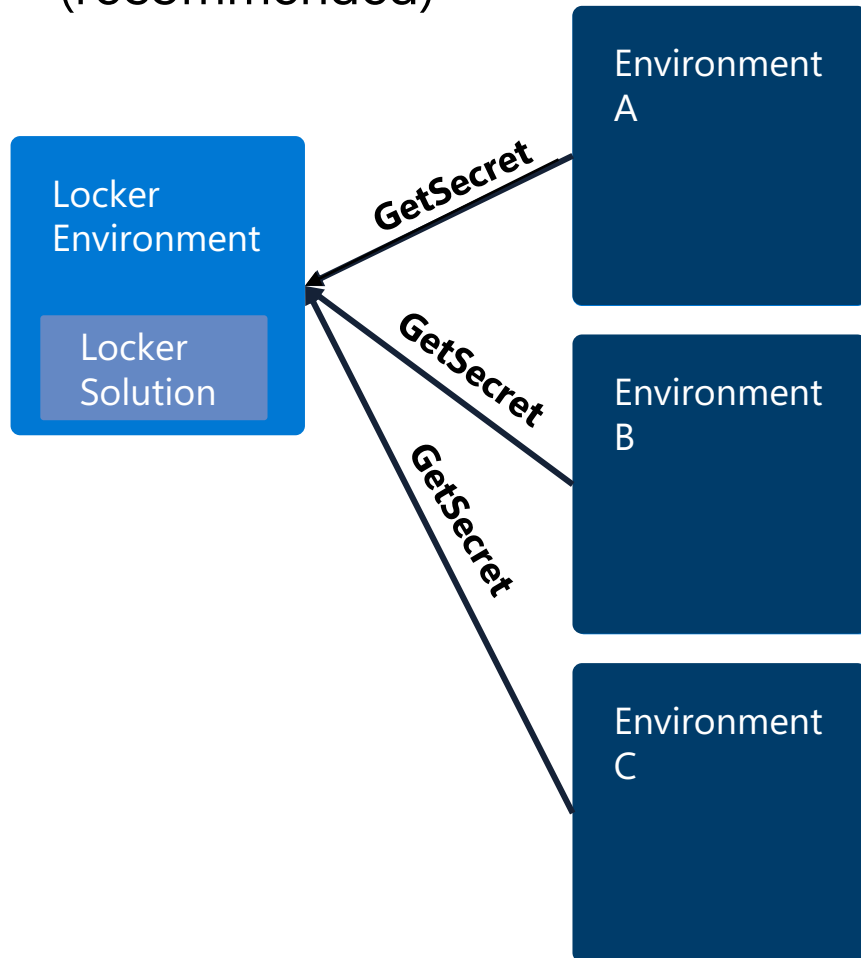
Agenda

- Installation
- Create Locker + Access Management
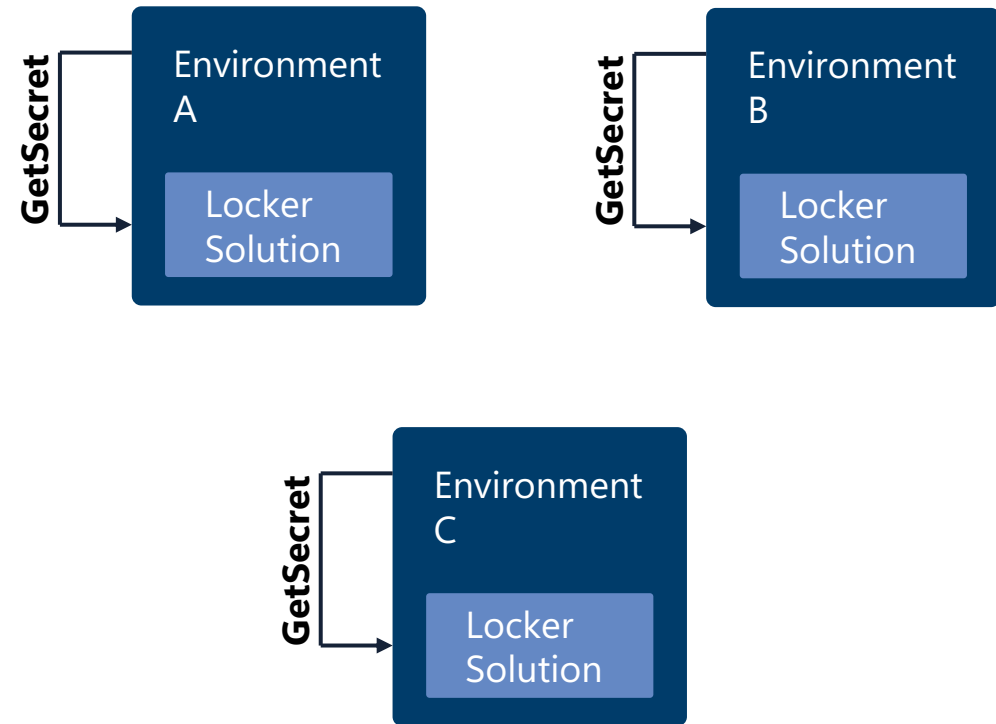- Create Values
- Use Lockers

Installation

Locker Environment Setup

- 1 shared Locker environment (recommended)

- Multiple Locker environments

Installation steps

1. Download managed solution file: GitHub - Locker Mngt

2. Import solution in Locker environment(s)

Assign Security Roles

The solution has 2 security roles. These roles must be assigned to all users and application users using the lockers according to needed privileges for each user.

Role	Create Locker	Edit Locker	Share Locker	Create Values	Edit Values	Read Actual Values
Locker Admin	✓	✓	✓	✗	✗	✗
Locker User	✗	✗	✗	✓	✓	✓
Locker Reader	✗	✗	✗	✗	✗	✓

*Only user with the security role "Locker User" will be able to see the actual value, otherwise these users will see the encrypted value.

Security roles and privileges - Power Platform | Microsoft Learn

Assign users/teams to Column Level Security profile

For any user to be able to see the actual value (Version) for a Value, the user has to be configured in the "column level security profile": **Locker Secrets Reader**.

When a user is not in this profile and even has full access to the record, the user will not be able to see the actual values. They will just see a password field (****).

Column-level security - Power Platform | Microsoft Learn

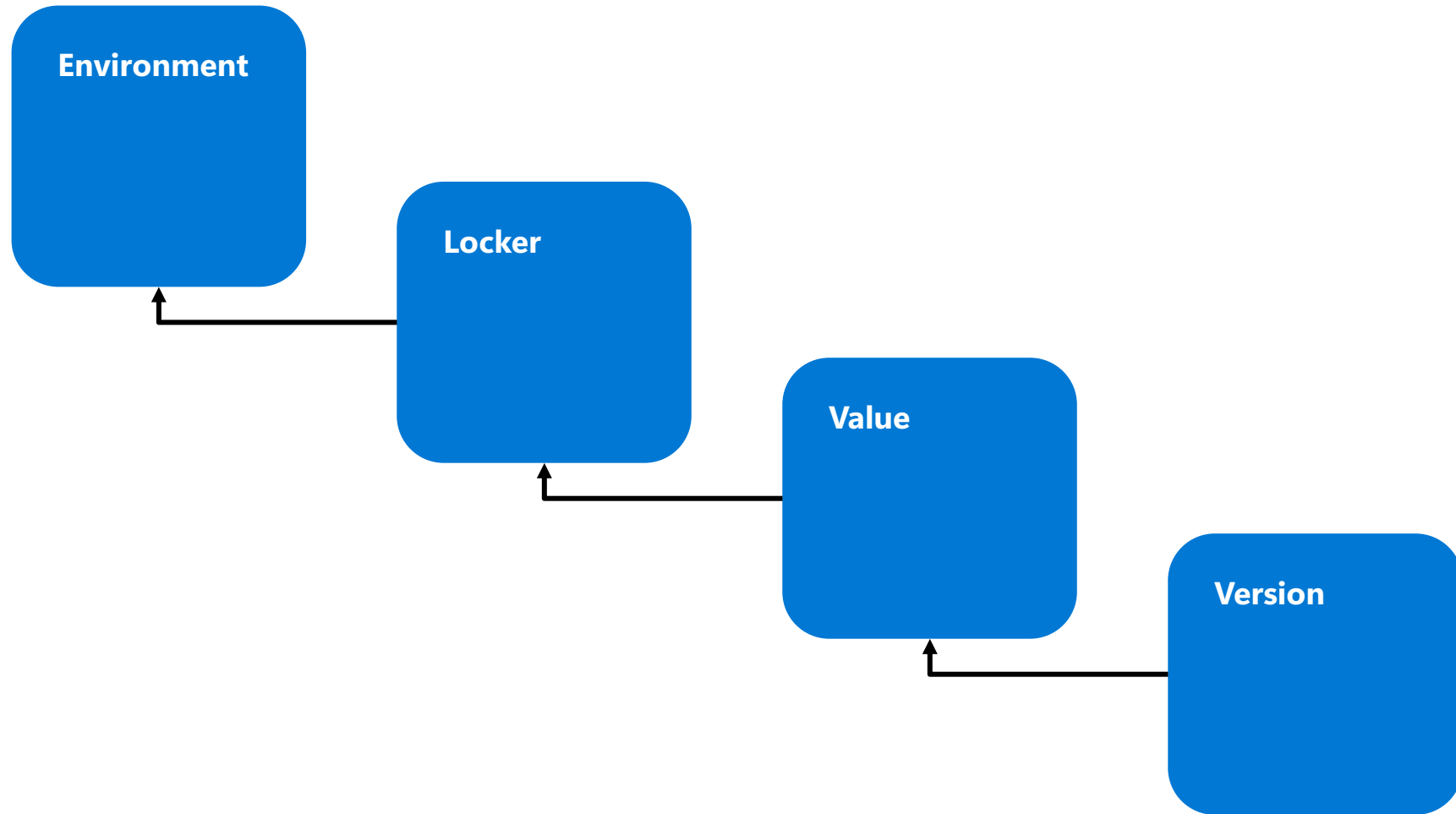# Assign users/teams to Column Level Security profile

1. Navigate to the Locker Management solution.

2. Select Column security profiles solution artifact.
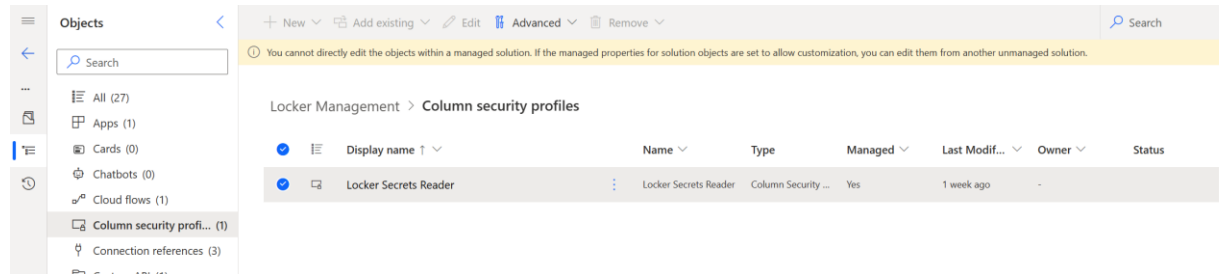
3. Select the Locker Secrets Reader profile.

4. Add your users, application users and teams on the profile.
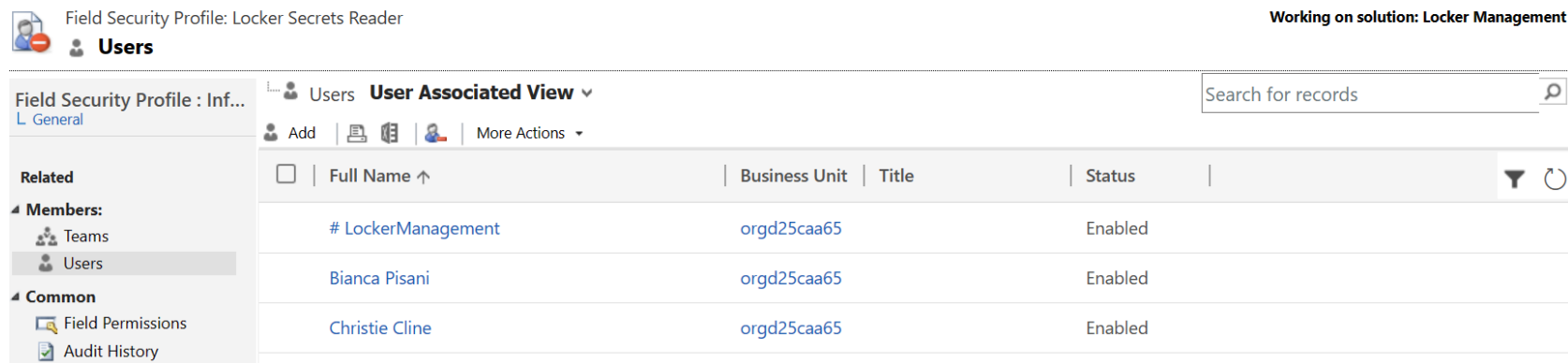
Data Tables

Create Locker

Locker

**Lockers** are created for a specific **environment** and has certain **values**. A locker is **owned by a user or a team** and is managed by a user with a Locker Admin security role.
The owner of the locker will grant access to other users or teams by **sharing** the locker.
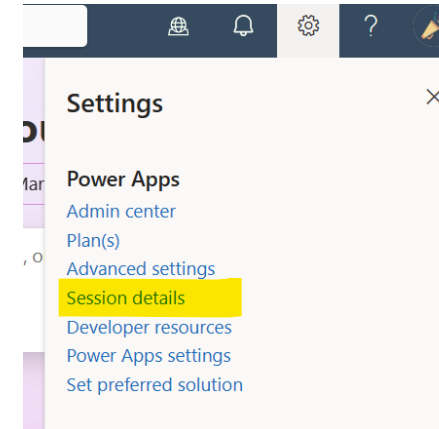
Example lockers:

A Locker can be created to keep track of values for a specific bot.

A locker can be used as a shared locker for several bots.
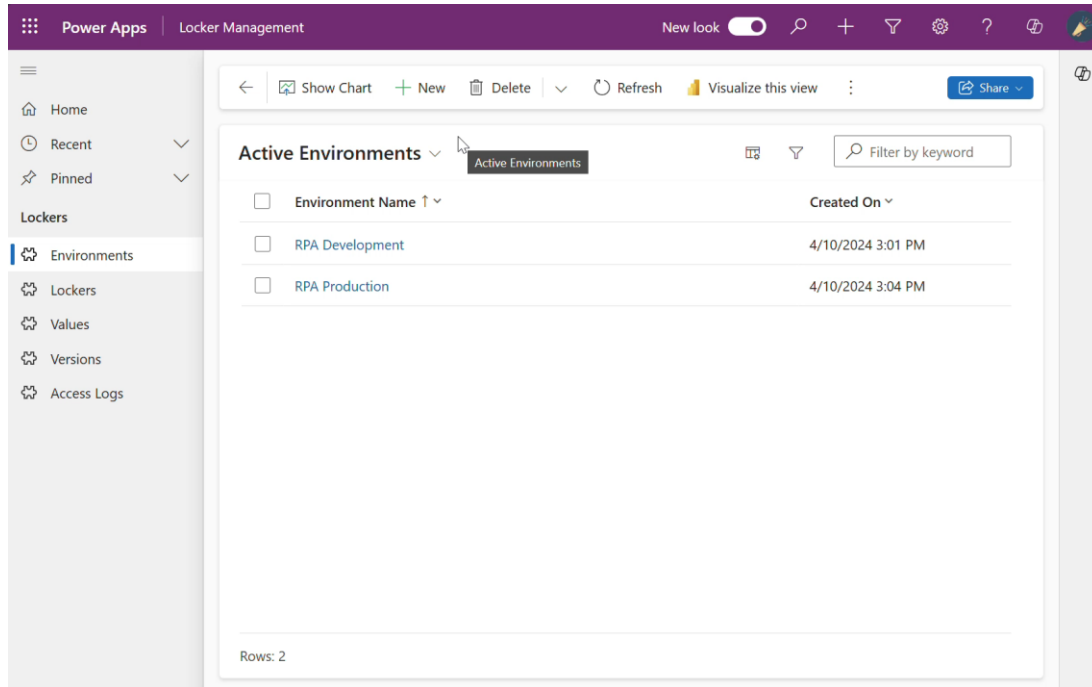
Create an Environment
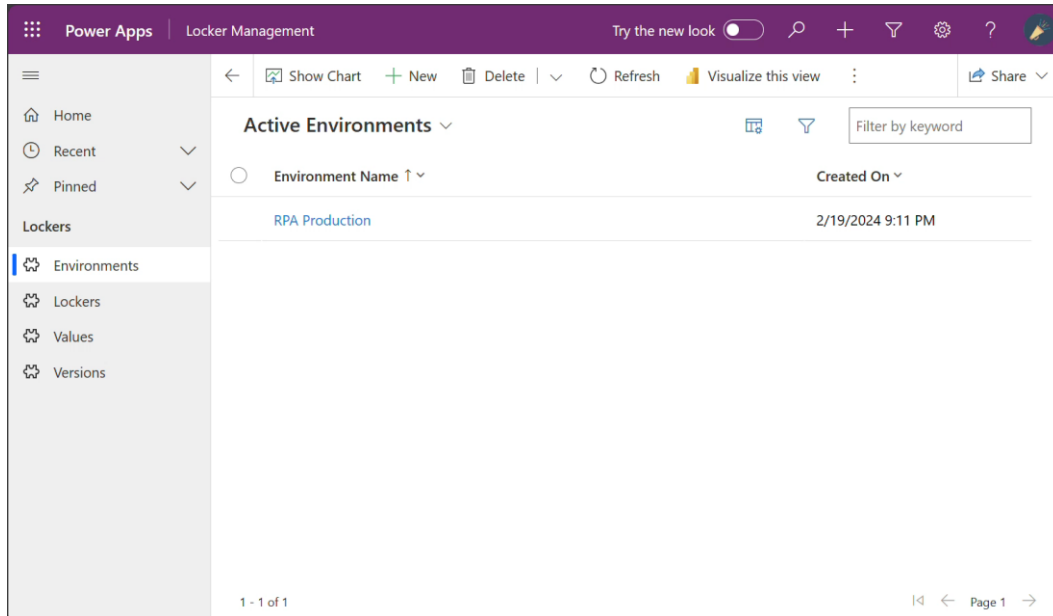
Get the Environment details from the settings – settings details in https://make.powerapps.com for the proper environment.

Or in the admin center: [Edit properties of an environment - Power Platform | Microsoft Learn](#)
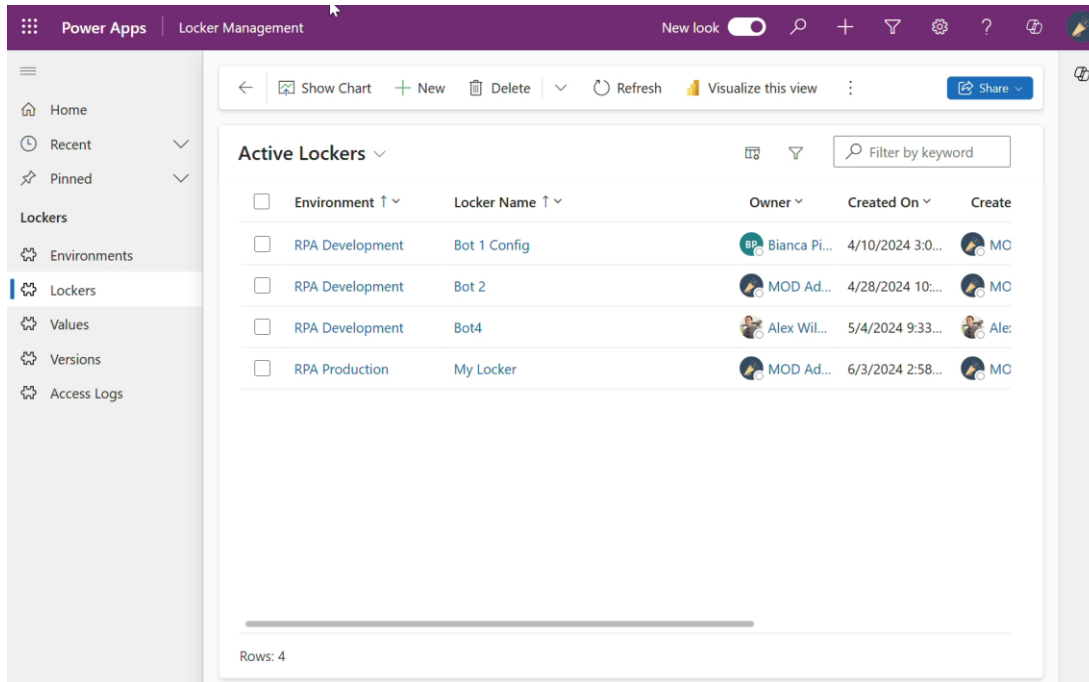
Create a Locker

A Locker is connected to an environment and has a name for the locker.

The name should be unique within the lockers for a specific environment.

The same locker name can be reused across different environments.
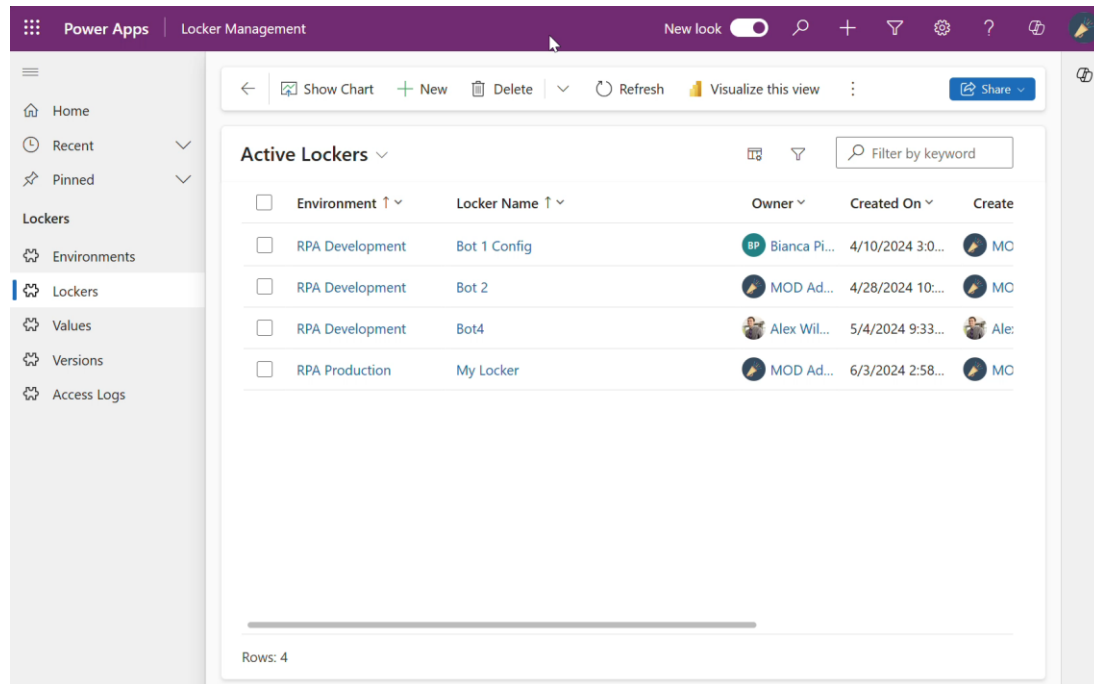
Access Management

Share a Locker

To give access to a locker for a user to be able create, edit or read values, a locker has to be shared with the user, application user and/or team.

Permissions to be used in the Sharing popup:

Review access

It is a best practice to occasionally review the access for each of the lockers.

In this example the locker has been shared with a user, an application user and a team.

Create Values

Values

Values can be created within a **Locker.** These are the names that will be used to retrieve the actual value for the Value in a Locker.
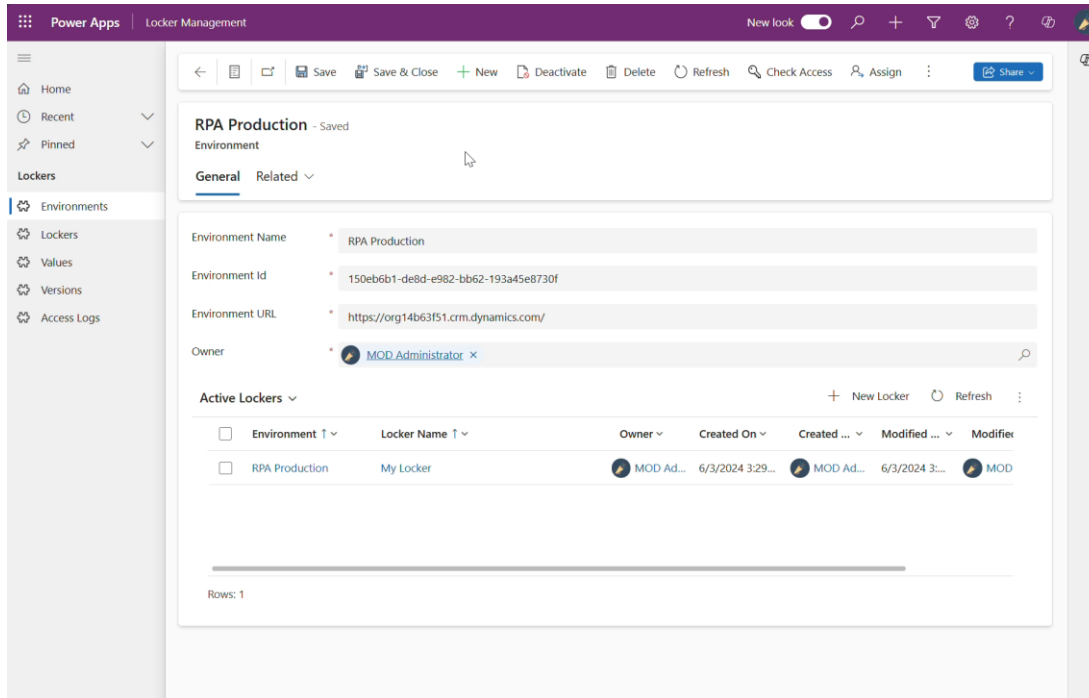
Example lockers:

Bot 1 username: to hold the actual value for the bot username.

Bot 1 password: to hold the actual value for the bot password.

SQL1ConnString: to hold the value of connection string to connect to SQL1.

Create Values

A Value is created for a locker. A Value is created from the locker record or in the Values list.

You can also set a Value to Private, which means this value is only accessible for the owner of the Value record and non-interactive users/application users.

Add a version for a Value

To add an actual value for a Value record, a version is created, which holds the actual value in an encrypted format. Only Locker users are able to see the secret values in a decrypted format.

Multiple versions can be created for one Value record. A version can have a start and/or end date.

If multiple versions would be valid, the most recent version will be used for the Value.

Versions can be created from the Value record or in the Versions list.

Versions can also be set to Private, same behavior as a private Value.

Use Lockers

How to read the actual value for a Value of a Locker

For Cloud Flows and Desktop Flows to retrieve the actual values from lockers the solution has a [Custom API](). This Custom API endpoint will get the correct Version for a Value in a Locker. The Custom API has 3 input parameters: Environment name, Locker name, Value and will return the actual Version.

The solution contains 2 sample flows, 1 example is a Cloud Flow which will retrieve a value. The other example is a Desktop Flow.

Cloud Flow

To get a secret (or actual version) for a Value you can use the unbound action of the Dataverse connector.

For the environment use the Locker environment.

For the Action parameters provide the proper values.

The connection should be created with an account (user or application user) which has access to the locker and the Locker User security role.

Desktop Flow

Invoke web service

Invokes a web service and stores the response text [More info](#)

URL: `%LockerInstanceUrl%/api/data/v9.2/rpacat_GetSecret`

Method: `POST`

Accept: `*/*`

Content type: `multipart/form-data; boundary=BoundaryData`

Custom headers:
```
Authorization: Bearer %LockerToken%
Content-Type: application/json
```

Request body:
```
{
"Environment": "RPA Production",
"Locker": "Bot 1 Credentials",
"Secret": "Bot 1 username"
}
```

Save response: `Get text into variable (for web pages)`

> Advanced

> Variables produced   GetSecretWSResponseHeaders   GetSecretWSResponse

GetSecretStatusCode

On error   Save   Cancel

To get a secret (or actual version) for a Value you can use the invoke web service action to call the custom API (rpacat_GetSecret).

For the URL use the Locker environment URL.

For the Action parameters provide the proper values.

To authenticate a proper Bearer token is required.

Desktop Flow (from PAD version 2.45)

To get a secret (or actual version) for a Value you can use Dataverse unbound action.

Select the locker environment name.

For the Action parameters provide the proper values.

To authenticate use a connection reference.

FAQ

Q: I can't create a Locker.
A: Assign the Locker Admin security role.

Q: I see '****' for the secret value in a Version record.
A: Add your user to the column level security profile.

Q: I see an encrypted string for the secret value in a Version record.
A: Add your user to the Locker User security role.

Q: I see a strange long string for the secret value in a Version record
A: Make sure you have the Locker User security role.

Q: I am not able to create a Value record.
A: Assign the locker User security role to your user account.

Microsoft Power Platform

Thank you!