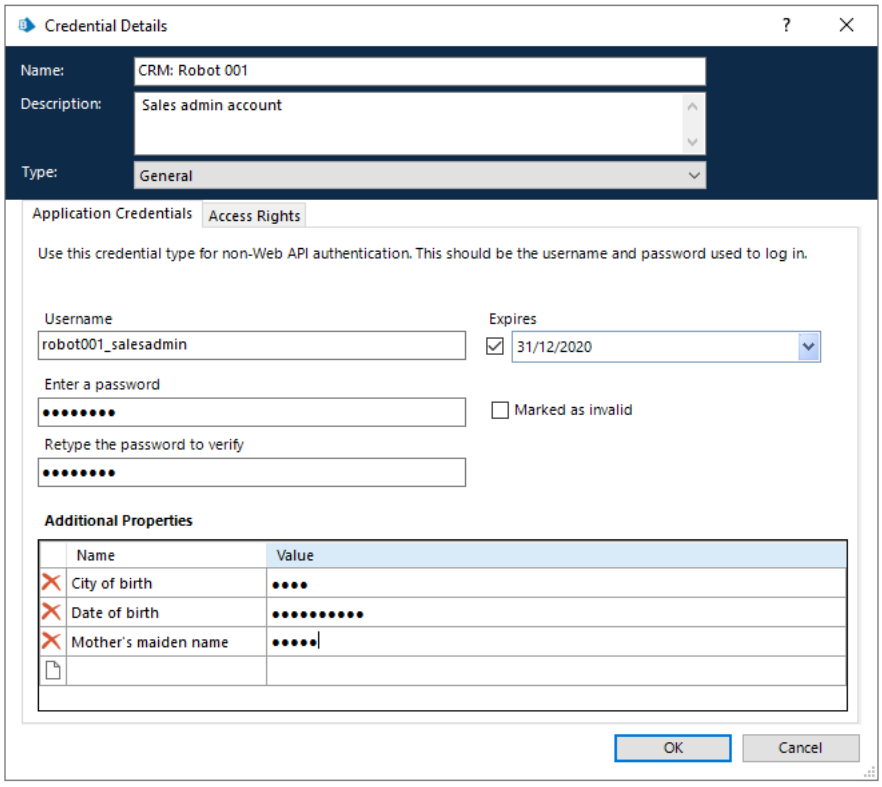
Credential Manager

The Blue Prism Credential Manager provides a secure repository for login details used to access target applications. By encrypting and storing credentials securely, Blue Prism can log into applications in a secure runtime environment while preventing the casual user or developer from re-using those credentials outside the production environment.

Credentials are stored in the Blue Prism database, but are encrypted in such a way that only those who are authorized can retrieve them. The encryption key is stored separately on the Blue Prism application server, and is used to provide credentials to validated clients.

The Credentials Manager determines which processes, runtime resources and user roles are able to access this information, and provides it on request if allowed by a set of permissions controlled via the System Manager.

Used in conjunction with Active Directory integration and the Blue Prism application server, the Credentials Manager creates a secure and fully audited access control capability for enterprise implementations.



Credential types

The different credential types reflect the supported authentication methods. Appropriate fields are available for the selected type.

- **General** – Used for non-Web API authentication.
- **Basic Authentication** – Used for basic web authentication to create the authentication header.
- **OAuth 2.0 (Client Credentials)** – Used for OAuth 2.0 web authentication using client credentials.
- **OAuth 2.0 (JWT Bearer Token**) – Used for OAuth 2.0 web authentication using JSON Web Tokens (JWT).
- **Bearer Token** – Used for Bearer token authentication.
- **Data Gateways Credential** – Used for Data Gateways configurations that require authenticated access to a database or HTTP endpoint.

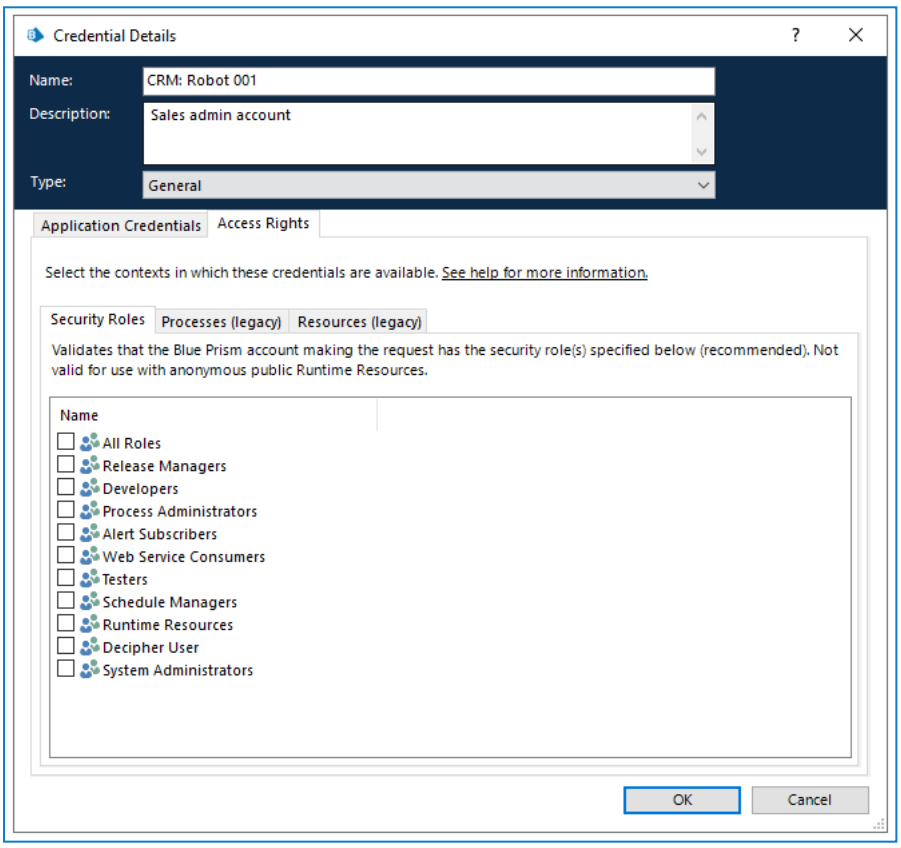
Authorization

Access to each set of credentials is governed at runtime by the following criteria:

- **Security Roles** – Validates that the Blue Prism account making the request has the security role(s) specified below (recommended).

  Not valid for use with anonymous public runtime resources.

- **Processes (legacy)** – Validates that the session identifier provided when requesting the credential relates to an active session and that the process specified within the session record is specified below (not recommended).
- **Resources (legacy)** – Validates that the session identifier provided when requesting the credential relates to an active session and that the resource specified within the session record is online and specified below (not recommended).



These restrictions work in combination, i.e. if a set of credentials is restricted by processes and resources then both of these criteria must be fulfilled to allow the credentials to be retrieved.

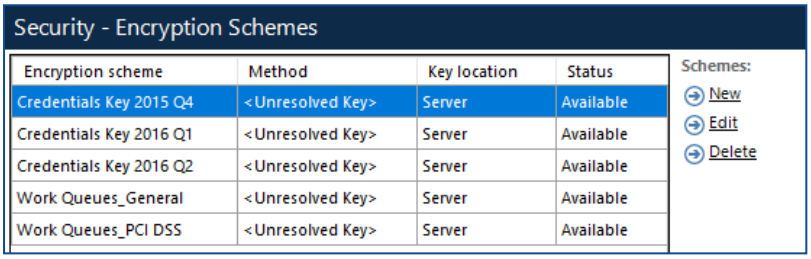
At the request of an authorized and validated client, a credential is decrypted locally on the Blue Prism Server and passed to that client via a secure connection. It is recommended that the access rights are configured so only user accounts with the appropriate security roles are able to access each credential. When a credential is limited by security role:

- The selected security roles must be held by the user(s) who require access to the credential when building or debugging a process or object.
- Only resources that are configured to explicitly authenticate against the environment will potentially be able to access the credential.
- Security roles must be held by the account used by resources to authenticate against the environment.

Recommended configuration

It is recommended that the configuration of the Blue Prism environment includes:

- Setting up encryption schemes with a server key location.
- Configuring the application server to store the keys in separate files and optionally applying custom security to the files to restrict access to the Blue Prism server service login account (and a named administrator).
- Configuring all clients to connect via an application server, and to establish secure connections.
- Leveraging single sign-on for Blue Prism.
- Ensuring an encrypted communication channel between application servers and the database.



Where application servers are not used within the environment, or where native communication security cannot be applied, it may be necessary to manually configure external security measures to prevent sensitive information from being transmitted as plain text.

Data security

Algorithm and key location

There are a number of encryption algorithms available which can be used to protect credentials and encrypted work queue information.

Name	Key length	Notes	Key generation information
AES-256 AesCryptoService (5.0.24+)	256-bit	Default implementation leveraging CBC	Blue Prism can be configured to use a manually generated key, or users can use the Generate Key functionality within Blue Prism. Keys generated within Blue Prism are created using RNGCryptoServiceProvider which provides a cryptographically strong sequence of random values.
AES-256 RijndaelManaged	256-bit	Default implementation leveraging CBC	
3DES	192-bit	CBC mode with keying option 1	

Protecting the key

When configuring an encryption scheme it is possible to select whether the key will be stored:

- **Database** - The encryption key will be stored within the Blue Prism database. This is commonly appropriate for scenarios where no application server is deployed.
  Supports clients that connect directly to the database, and those that connect via an application server.
- **Blue Prism Server** (recommended) - The encryption key will be stored on the application server. In this scenario, the key will need to be manually deployed to each application server in the environment. This is the most commonly selected scenario as it ensures the key is stored separately to the encrypted data.
  Supports clients that connect via an application server.

  When selecting to store the key on the application server it can be stored in the Blue Prism configuration file, or in a separate Blue Prism managed file. By selecting to use a separate file it is possible to add custom controls such as applying EFS to restrict access.
  If restricting access, it is necessary to ensure access is provided to the Blue Prism server service account as a minimum.

Data encryption/decryption

When a client device submits data that needs to be stored using reversible encryption, or requests data that is stored using reversible encryption, the device that is responsible for carrying out the conversion between plain-text and cipher-text will be dependent on how the client device is connected to the environment.

- **Application server connection** (recommended) - The application server is responsible for converting between plain-text and cipher-text for client devices that connect via a Blue Prism application server. When appropriately configured, the plain-text will be transmitted between client and server over a secure channel, and the cipher-text will be transmitted between the server and the database over a secure channel.
- **Direct database connection** (not recommended) - Client devices that have a direct connection to the Blue Prism database will be responsible for requesting the key and locally converting the data item between plain-text and cipher-text.
  When appropriately configured the cipher-text and key will be transmitted between the client and database over a secure channel.

Irrespective of where the conversion takes place, once the conversion has taken place the memory on the device is immediately cleared and disposed.

Key revocation

Blue Prism provides the ability to easily revoke a key, and there is an option to forcibly revoke a key (i.e. immediately convert all data encrypted with an old key to use a new key).

The steps required to configure Blue Prism to use a new key for all subsequent data encryption and decryption include:

1. Create a new encryption scheme record.
   Where the key is stored in the database, add the key to the record.
   Where the key is stored on the application server, update the configuration of each application server to hold the key.
2. Update the Credential Manager to use the new scheme.
3. Update any applicable work queues to use the new scheme.
4. Mark the “old” encryption scheme as unavailable.

Data that is encrypted with the previous keys will still be decrypted using those original keys, but when that data is updated it will be re-encrypted with the currently active key(s).
To forcibly update all data encrypted with previous keys the AutomateC.exe command line switch /reencryptdata can be used (optionally specifying the batch size and maximum iterations - see the Product Help for more information.)

Frequently Asked Questions

How is the key passed to the client?

At the request of an authorized and validated client, a credential is decrypted on the Blue Prism Server and the plain-text result is passed to the client via a secure connection.
The keys are NOT shared with the client unless they are configured to use a direct database connection (not recommended). In this scenario the key will be shared over a secure channel.

Is it possible to periodically use a new key for data encryption?

Yes. An overview of the steps required to configure the platform to use a different key for data encryption are included within the main body of the document.