



Progeny Security Features

Your Data is Reliably Secure and Easy to Manage

User Identification and Authentication

The User ID and encrypted password control who has access to a database.

LDAP Integrated Login

Integrate your LDAP user accounts with User IDs in Progeny for an added layer of network security and control.

Auditing

With two forms of auditing available, the Chain of Custody audit maintains a record of transactions performed on the database, while the Field audit records the values that are updated, displaying old and new values for every modification. Both audits time stamp each event with a date, time and User ID. In addition to these features, which are both available through the user interface, the database engine has extensive auditing available in the form of log file operations.

Database Encryption

Database encryption features allow you to choose the level of database encryption. You can choose to secure your database either with simple encryption, or with strong encryption. Simple encryption is equivalent to obfuscation. Strong encryption renders the database completely inaccessible without an encryption key.

Transport-Layer Security

You can use transport-layer security to authenticate communications between client applications and the database server. Transport-layer security uses simple, elliptic curve, or RSA encryption technology. Separately licensable components are required for RSA and elliptic curve encryption.

Discretionary Access Control Features

The actions that a user can carry out while connected to a database are defined within the User ID.

User and User-Class Definitions

Users are assigned to class definitions to determine the roles and permissions throughout the software. General permissions such as access to spreadsheets, the data warehouse, markers, and form design are also assigned to the User ID. Integrated login is available for environments where it is required.

Field Level Security

Class based security can be defined even at the field level. Roles and permissions can be setup to determine what rights a given category of users has to modify or view data fields.

Folder Level Security

The ability to read, add, delete, and modify is controlled at the folder level. Security applies to folders throughout the software including data field folders. This flexibility is ideal for setting up scenarios where one group of users is not able to read or to modify another group's data. Folder level security makes it possible for several institutions or groups to share a single database. Each institution may be unaware of the other members.

Pedigree Level Security

Read and write access can be assigned to an individual pedigree. The permissions are assigned via the user class.

Database Level Security

The user can set the level of security on the database by restricting functions such as the ability to add, delete, or modify a pedigree or individual folder as well as the ability to add, delete, or modify a data folder.

Password Expiration

Passwords have a forced expiration every 180 days after which they must be changed. The new password is validated to not contain the old password.

Password Naming Conventions Enforced

Passwords are required to be 6 characters long, contain both letters and numbers, and cannot include the User ID.

External Access Protection

Access to the Progeny database from outside the Progeny application is restricted to the Progeny administrator User ID.

Login Auditing

Each login to the database is audited. The User ID, operating system, application, host name, login date, and logout date are recorded for every successful and unsuccessful login.

C2 Certification

C2 is a set of security guidelines established by the U.S. government to maintain consistency within their organization. If you have the appropriate hardware, you can set up your machine to run in a C2 certified manner.

Web Security Features

Progeny Web Application depends on and builds on top of the security features available in the desktop application. To protect the application against web specific security vulnerabilities, a security layer was added with the following protections

Server Generated User Tokens Have Expiry Time

Once the user is logged on, user credentials are kept in a user session object on the server and are not to be transported across the wire with every request. Instead, leased user tokens are used to identify the user each time a request is transmitted between client and server. The tokens have the following properties:

- a. They expire after a set period of time, at which point user will be leased another secure token. Any attempts to logon using an expired token will be rejected even if all the other credentials are correct.
- b. They are mapped to the actual user credentials on the server before accessing the database.
- c. They are generated in a random and secure manner to ensure that they cannot be guessed easily

SQL Injection Attack Protections

Progeny Web uses best practices for avoiding SQL injection attacks. Any parameters received from the client are escaped in a database specific manner before they are passed to the database. That way the parameters are treated as literal values instead of being treated as SQL code when the query is executed by the database engine. Malicious database code embedded inside the query is deactivated.

Treat All Requests from Internet as Untrusted

Progeny Web treats each request received from the Internet as one that cannot be trusted. In that vein, each request is checked to ensure that the user has access to the resources that they say they are requesting before granting them access. For instance if an attacker intercepts a request from a real user requesting pedigrees from a certain folder and decides to change the request so that it returns a list of pedigrees from another folder that the user would normally not have access to, the Progeny Web security layer will reject such a request since it first checks that the user does have access to the folder being requested.

SSL Encryption of All Request Progeny Web Application

We strongly encourage our customers to locate Progeny Web on a server that has an SSL certificate and to ensure that all HTTP requests are encrypted using the SSL/TLS mechanism. This prevents Man In The Middle attacks by hiding the real meaning of the data in the requests.

Cross Site Scripting Protection

Progeny Web goes a long way in avoiding cross-site scripting attacks. It does this by ensuring that any data that comes from the user and may eventually be displayed in the application as HTML is escaped. This ensures that any malicious JavaScript and/or other HTML code that was entered by the user is displayed as a literal string instead of being potentially executed by the browser and causing unspecified behavior.

Strong Public Cryptographic Algorithms

Only publicly available cryptographic algorithms are used to encrypt data. Published algorithms tend to be strong because they are well tested and hence most if not all vulnerabilities would have been discovered and addressed.

Measures Against Session Stealing

Progeny employs the following measures to ensure that stolen sessions cannot be used to attack and possibly steal and/or decimate the application data.

- a. Make user sessions expire after set period of inactivity
- b. Allow users to invalidate their sessions when done using Progeny Web by providing LOGOUT facility
- c. Use Servlet Filters to inspect each request for validity of source (client) IP address
- d. Require that every request to the server be SSL encrypted

Database Behind Firewall

We strongly encourage our users to locate the Progeny Database behind a firewall. This ensures that any attack on the web server does not necessary result in a direct compromising of the database since there will be another security layer for the attacker to get past.