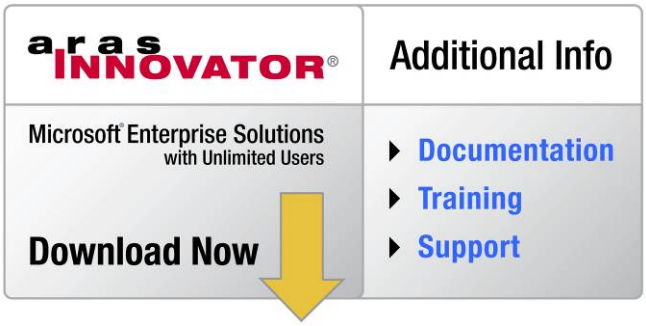
Aras Innovator
Login Security

Aras Innovator 9.2

Document #: 9.2.003302010

Last Modified: *4/1/2010*



ARAS CORPORATION

Aras Corporation
300 Brickstone Square
Suite 904
Andover, MA 01810

Phone: 978-691-8900
Fax: 978-794-9826

E-mail: Support@aras.com
Website: http://www.aras.com

Table of Contents

Send Us Your Comments

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

- **Email:**
  Support@aras.com
  Subject: Aras Innovator Documentation

Or,

- **Postal service:**
  Aras Corporation
  300 Brickstone Square
  Suite 904
  Andover, MA 01810
  Attention: Aras Innovator Documentation

Or,

- **FAX:**
  978-794-9826
  Attn: Aras Innovator Documentation

If you would like a reply, please provide your name, email address, address, and telephone number.

If you have usage issues with the software, please visit http://www.aras.com/support/

Overview

Aras Innovator provides the flexibility to allow administrators many options when controlling the maintenance of user logins to Aras Innovator. This document will concentrate on the Aras Innovator Security feature, but this is not the limit of the possible configurations. Aras Innovator has many internal security features that can be enabled to maintain control of user password, and session expiration. Alternately, the Aras Innovator client logon may be customized through the use of logon hooks that can allow alternate login methods like Windows Authentication. For details on this feature, subscribers can access the *Aras Innovator – Windows Authentication Setup* document from the Documentation folder of the Aras Innovator CD Image.

Changes outlined in this document should not be made to a production instance of Aras Innovator while it is running. Please plan to implement these features only when users are not connected to the system, in a controlled deployment.

1 Aras Innovator Security Features

Aras Innovator Security is a set of features that allows the administrator to control actions associated with user authentication like password restrictions, session timeout, and account expiration. These features are only intended for use with users who are authenticated using Aras Innovator, and not alternate methods like Windows Authentication. Some of these features directly conflict with other authentication methods. Also, it is recommended that logins used for purposes like the Aras Innovator Service should be excluded from these features where possible, as these users will be unable to control authentication without administrator intervention.

1.1 Session Expiration

Aras Innovator has the ability to require users to re-authenticate themselves after a session has timed out. By default, users never have to re-authenticate once they have logged in, but with this feature you can require all timed out sessions to do so. The changes that must be made to implement this feature only apply to the Innovator Server instance that the user is connecting to. If one database is connected to two Innovator Server instances, both must be configured, if you want both to use this feature.

First, you will need to set the session timeout to the Innovator Server. In the installation folder, edit the \Innovator\Server\web.config file. Under sessionState, set the timeout value to a positive integer in minutes. This is the number of minutes any session can go idle until timing out.

```
<sessionState
        mode="InProc"
        stateConnectionString="tcpip=127.0.0.1:42424"
        sqlConnectionString="data source=127.0.0.1;Trusted_Connection=yes"
        cookieless="false"
        timeout="480" />
```

Second, you will need to enable the session time-out checking on the Innovator Server. To do this, edit the InnovatorServerConfig.xml in the root of the installation folder and add the following tag:

```
  <operating_parameter key="enable_session_time_out" value="true"/>
```

If you need to disable the session time-out checking, simply set this value to “false”.

After making these changes you must increment the CustomBuildVersion by 1

```
<Innovator>
  <UI-Tailoring login_logo="...
  <IEClient RequiredSyncMode5="Never" CustomBuildVersion="2" />
  <operating_parameter key="debug_log_flag" value="false" />
...
```

Finally, restart the World Wide Web Publishing service on the server to ensure the server cache is refreshed.

1.2 Blocking Sessions that Fail to Authenticate

Aras Innovator has the ability to block failed attempts to authenticate. This feature is especially useful if Aras Innovator has a public URL that may be the target of automated attempts to login. When any client, identified by IP address, fails to connect in a specified number of tries, the client will be blocked from connecting for a specified number of minutes. There are two variables that must be set to enable session blocking.

AccountLockoutThreshold_triesNum – This defines the number of tries a client has to authenticate before being locked out. The value should be set to a positive integer. The default value is set to -1, so that this parameter is ignored until set by the administrators.

AccountLockoutDuration_minutes – This defines the number of minutes a client will be locked out before being allowed to attempt to connect again. The value should be set to a positive integer. The default value is set to -1, so that this parameter is ignored until set by the administrators.

1.3 Password Restrictions

There are several features that control password restrictions of Aras Innovator Users.

1.3.1 Password Format

There are two variables that control password format. To edit these Variables, select Administration\Variables from the TOC.

User_pwd_symbols_min_number – This variable controls the minimum number of characters a password must contain, in total. The value should be set to a positive integer. The default value is set to -1, so that this parameter is ignored until set by the administrators. This value will not be enforced on current passwords until they are changed, all new passwords will use this variable.

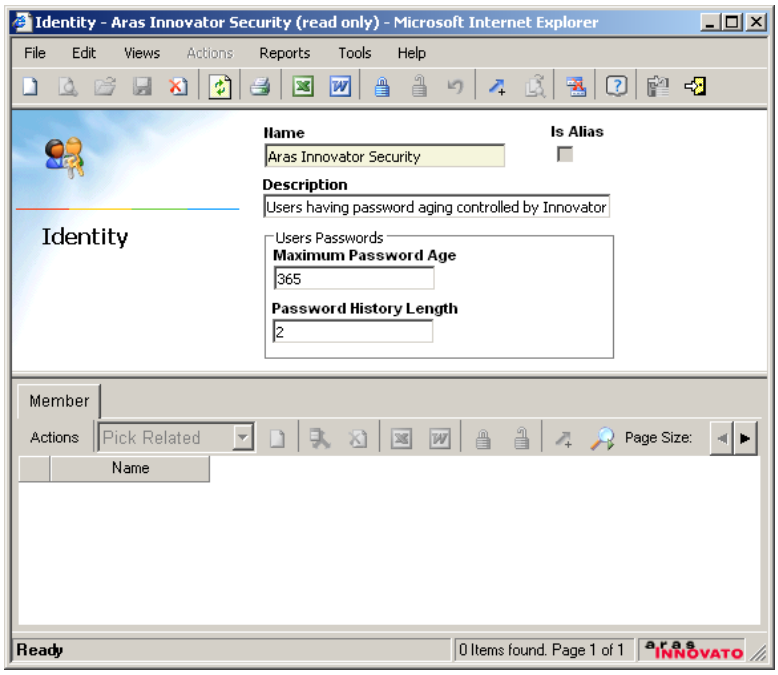
User_pwd_digits_min_number – This variable controls the minimum number of numerical characters a password must contain. The value should be set to a positive integer. The default value is set to -1, so that this parameter is ignored until set by the administrators. This value will not be enforced on current passwords until they are changed, all new passwords will use this variable.

As an example, if User_pwd_symbols_min_number=6 and User_pwd_digits_min_number=1, then all passwords should be at least 6 characters long and contain at least one digit. “h3lloo” would be acceptable for this set of restrictions.

1.3.2 Password Expiration

There are two properties that control password expiration. To edit these properties, select Administration\Identities from the TOC. Every Identity has the ability to control password expiration, but it is doubtful most installations will need this level of control. Also, administrators want to be sure to exclude any system users that might be used to allow things like the Aras Innovator Service to connect to Aras Innovator. These system users will not be able to change a password without administrative intervention. Unless administration has a specific schema in mind, Aras recommends creating an identity to

manage password aging of users, and making User Identities members of this “Aras Innovator Security” Identity to manage password aging in one location.



Maximum Password Age – This is the maximum number of days a user may use the same password, before they will be prompted to change their password on login. This value is blank by default, but should be set to a positive integer.

Password History Length – This is the number of past password the system will remember. Users may not reuse any password already in the password history. This value is blank by default, but should be set to a positive integer.

1.4 Account Inactivity Report

There is a report included to determine what accounts are inactive in Aras Innovator. Administrators can use this report to determine if any accounts should be disabled based on inactivity. To access this report, select Administration\Users from the TOC. Then, select Reports→Inactive Accounts from the main menu.

2 Securing built-in Aras Innovator Accounts

The core Aras Innovator database comes with 3 built-in accounts. These are “Innovator Admin” (username=admin), “Super User” (username=root), and “Vault Admin” (username=vadmin).

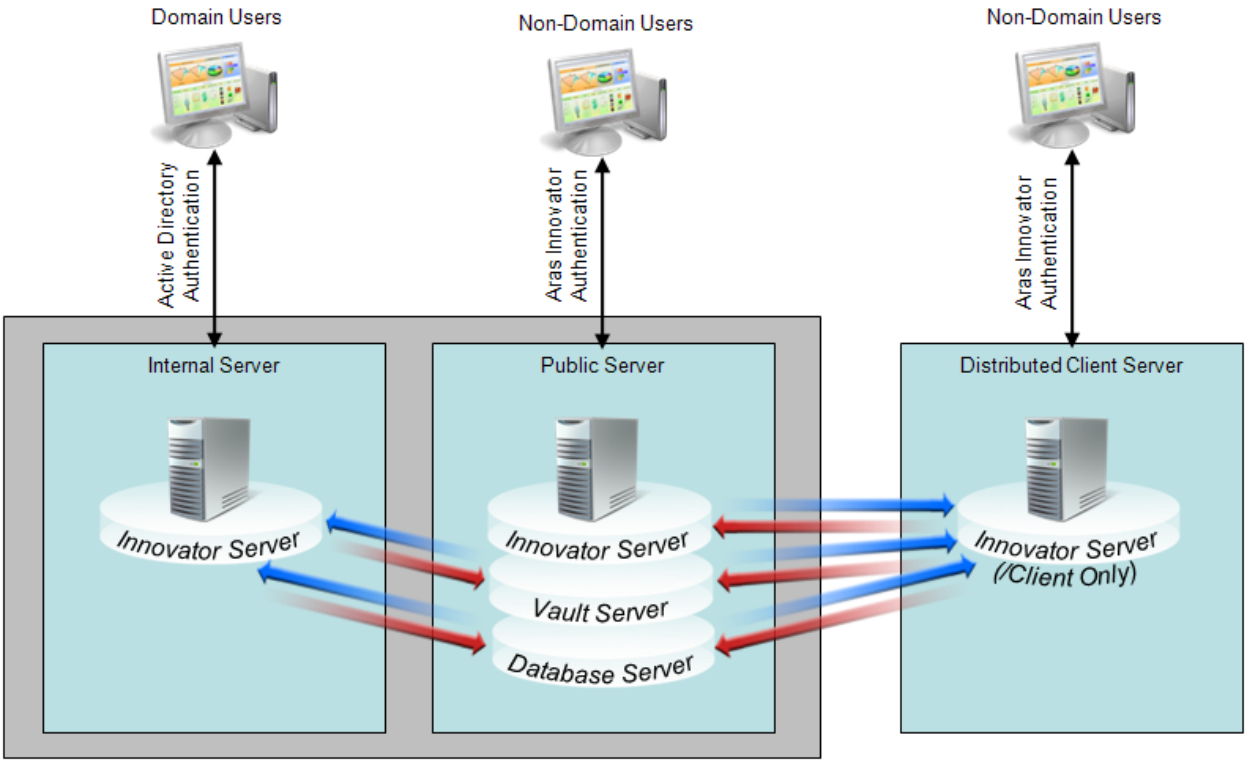
The Innovator Admin and Super User accounts should be changed to prevent them being used by persons who know something about the default values of these passwords by disabling these accounts and only enabling logon during periods controlled by strict configuration management principals. Users should be made members of the Administrators Identity to have administrative privileges assigned on their own account, rather than using the Innovator Admin or Super User accounts.

The Vault Admin user cannot be disabled if the VaultServer feature of Aras Innovator is being used. The best way to restrict access to this account is to generate a random, sufficiently long password as to be astronomically improbable to guess, and to store this password in encrypted form in the VaultServerConfig.xml file.

3 Mixing Authentication Methods

Aras Innovator allows for a flexible set of configurations for authentication, and for site structure. By combining the ability to distribute the different tiers of innovator with the different authentication modes, administrators can create a deployment that leverages more than one authentication method.

The following is an example of an existing production deployment of multi-tier mixed authentication control.



In this diagram we have three servers running Aras Innovator.

Public Server - This server represents the main instance of Aras Innovator. This server runs the Innovator Server, Database Server, and Vault Server tiers of Aras Innovator. The URL for this server would be used by internal and external users of Aras Innovator, and would deploy the Aras Innovator Security features. Users would authenticate against this server using standard Aras Innovator authentication methods.

Internal Server – This server represents a second instance of Aras Innovator on the same network as the Public Server, but does not stand alone. This server only consists of the Innovator Server tier of Aras Innovator, and would refer back to the Public Server when calling the Database Server tier or the Vault Server tier. The URL for this server would be

used by internal and external users of Aras Innovator, and would deploy the logon hooks. Users would authenticate against this server using Active Directory, and would not be subject to the session timeout restrictions of the Aras Innovator Security feature of the Public Server.

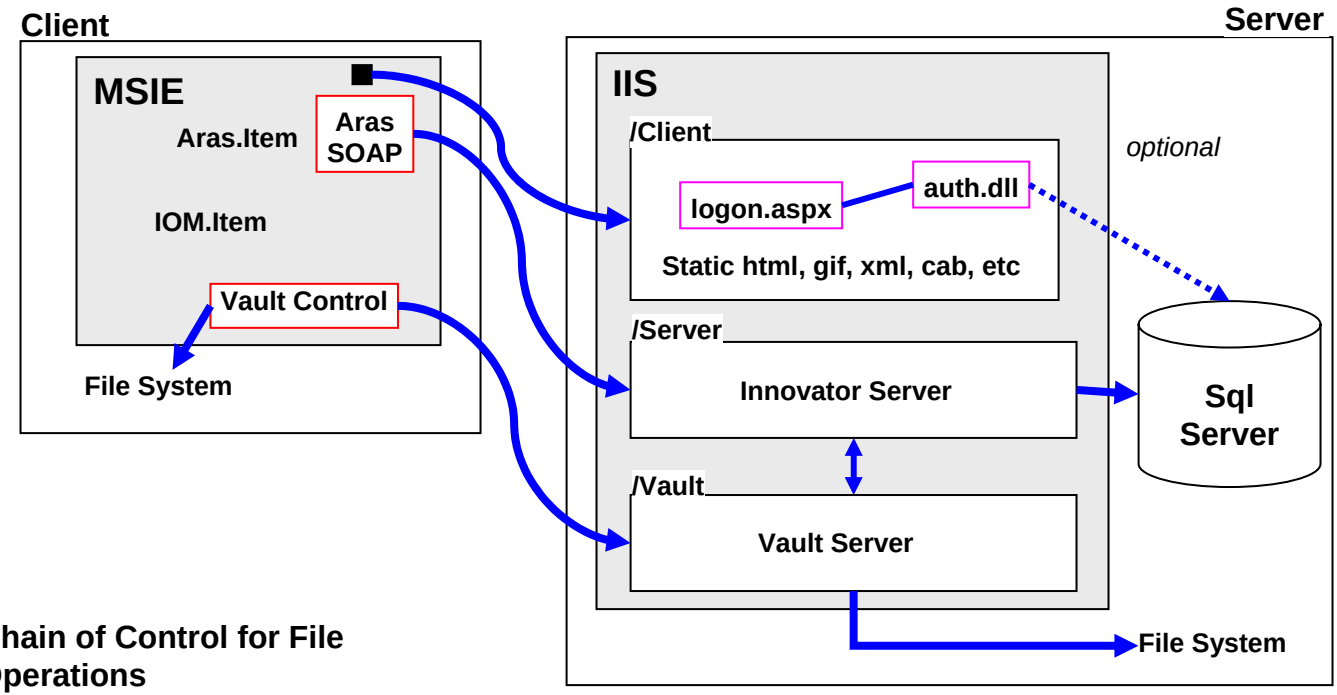
Distributed Client Server - This server represents an instance of Aras Innovator deployed on a different LAN than the Public Server. This server only consists of the Innovator Sever tier of Aras Innovator. Furthermore, only the /Client folder would be used from this tier. When calling the /Server folder, all requests would be redirected to the Public Server. As with the Internal Server, all requests would refer back to the Public Server when calling the Database Server tier or the Vault Server tier. This deployment is for two reasons. One, all calls to the /Client folder, like for UI images, would be done on a local network at the remote site, and would help performance over a slow WAN connection. And two, this deployment allows the Public Server to control the Authentication and session management, including Aras Innovator Security features like session timeout.

While this example only represents one configuration possibility, it does represent the flexibility of access control that can be achieved with the Aras Innovator platform.

4 Reference Diagrams

4.1 Architecture

Aras Innovator Architecture



Chain of Control for File Operations