NT Authentication Configuration Guide

Version 11
Last Updated: March 2014

Overview of Ad Hoc Security Models

Every Ad Hoc instance relies on a security model to determine the authentication process for a user and the authorizations granted to the user. Ultimately Ad Hoc must be able to determine the user, their role(s) and which organization the user is associated with, regardless of the security model chosen.

Following are brief descriptions of the various security models.

Standard – as the default security model, credentials supplied in a login page are verified by interrogating the Ad Hoc metadata database associated with the instance. Users, organizations and roles must have been created by the System Administrator and are managed by the System Administrator.

Database – similar to the Standard security model, the Database model relies upon a database external to Ad Hoc to verify the logon credentials. Users are managed outside of Ad Hoc. Organizations and roles are managed within Ad Hoc.

Session – the user is authenticated elsewhere, typically a parent application, and the user name is passed to Ad Hoc. Roles and organizations are managed within Ad Hoc.

NT Authentication – the user is authenticated by the Windows operating system. Roles and organizations are managed within Ad Hoc. User/Role associations are adjusted by comparing the User/NT User Group to the User/Role(s) in the Ad Hoc metadata. This is one of the single sign-on security models.

SecureKey – the user is authenticated by a parent application. Access to Ad Hoc is controlled by determining if the IP address of the caller and the “approved” IP addresses match or grant access. Roles and organizations must be managed within Ad Hoc. User/Roles are adjusted to match the information provided by the parent application. This is one of the single sign-on security models.

NT Authentication

NT Authentication is one of the single sign-on security models supported by Ad Hoc. In this model, Ad Hoc relies on the Windows operating system to authenticate the user and identify the NT User Groups to which the user belongs. The NT User Groups are matched to existing roles defined in the Ad Hoc instance to determine what databases and features the user is authorized to use.

This guide provides assistance for system administrators that want to use NT authentication and authorization with the application. Following are the configuration steps required:

1. Create and configure roles within Ad Hoc.
2. Update *_Settings.lgx* to use NT authentication.
3. Disable anonymous access for the application virtual directory.
4. Set the *ahUserGroupID* session variable to the ID of a valid application user group.

Configure roles within the application

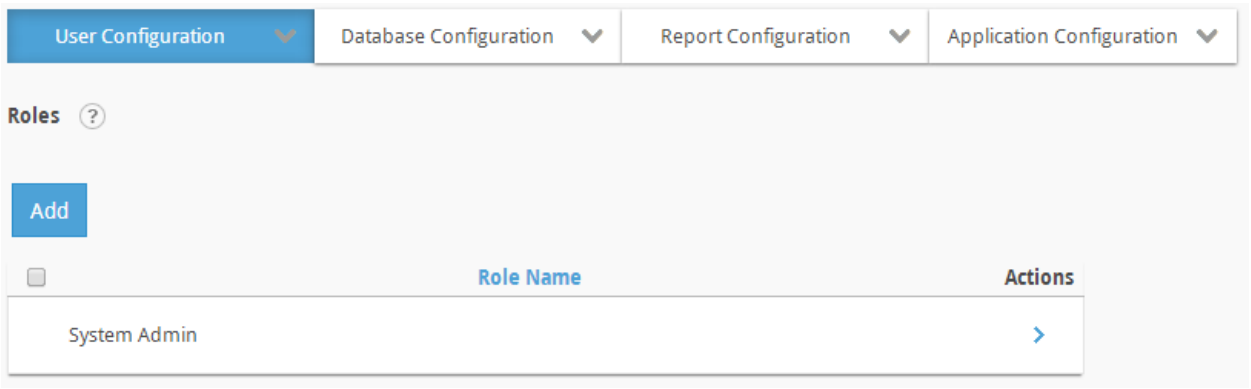
During the login process with NT Authentication, Ad Hoc determines the NT User Groups for the user and matches them with the Roles defined in Ad Hoc. Consequently, the Roles should have been defined prior to implementing the NT Authentication security model. Make a list of the NT User Groups that require access to Ad Hoc and use the Ad Hoc interface to create matching Roles.

Note:
To avoid confusion, "NT User Groups" are equivalent to "Roles" in Ad Hoc. "Organizations" in Ad Hoc are independent of the "NT User Groups". The System Administrator may still organize users within Ad Hoc into organizations similar to the "NT User Group" designations, but it is not required.

Note:
A user needs to have at least one Role matching one NT User Group exactly to gain access to Ad Hoc. Not all NT User Groups have to be replicated as Roles in Ad Hoc.

To create roles in the application:

1. Login to Ad Hoc as an administrator.
2. Click **Configuration** to enter the configuration page.
3. Hover over the **User Configuration** tab and click on **Roles** to access the role configuration screen.
4. Click **Add** to begin creating a new role.
5. Type a name for the role and configure the access rights. The name of the role must exactly match the name of the Windows NT User Group.
6. Click **Save** to commit the changes.
7. Repeat steps 4 through 6 for any additional Windows NT User Groups.



To create an "administrator" role in the application:

1. Login to Ad Hoc as an administrator.
2. Click **Configuration** to enter the configuration page.
3. Hover over the **User Configuration** tab and click on **Roles** to access the role configuration screen.
4. Click **Add** to begin creating a new role.
5. Type ***adHocAdmin*** as the role name. Assign Administration permissions and all databases to the role.
6. Click **Save** to commit all the changes.

Notes:

1. *Give users administrative privileges by creating a Windows NT user group called **adHocAdmin**. Members of the NT **adHocAdmin** group have the same access rights as users assigned to the **System Admin** role in Ad Hoc.*
2. *Ad Hoc always maintains a list of organizations, users and roles. If using NT authentication, it is not necessary to create user accounts within the application. User accounts are automatically created when a new user logs in for the first time.*
3. *The initial default username for the application is admin and the password is password. Administrators should change the default password after installing*

the application.

Update the *_Settings.lgx* file to use NT authentication

The application uses the *_Settings.lgx* file to obtain a variety of configuration information, including security parameters.

The *_Settings.lgx* file is located in the _Definitions folder under the root folder specified during installation (e.g., *C:\Program Files\Ad Hoc_Definitions*). To update the *_Settings.lgx* file for NT Authentication:

1. Make a backup of this file
2. Open the file in a text editor (e.g., Notepad)
3. Replace the content of the <Security> element with the following code:

```
  <Security NTAuthenticationDomain="<domain_name>" SecurityEnabled="True"
AuthenticationSource="AuthNT" LogonFailPage="LoginError.aspx" />
```

4. Modify the NTAuthenticationDomain attribute value. Organizations running a domain-based network must include the **NTAuthenticationDomain** attribute to the **Security** element. Replace **<domain_name>** with the name of the company's domain (e.g., LogiXML).
5. Save the file

NTAuthenticationDomain. For NT Authentication, you may set this value to name of the Domain that is authenticating users. The system will only accept users authenticated from that domain, and the domain name will be removed from the user name. So username "MyDomain\Username" becomes simply "Username". Also, when working with the UserRoles.NT element, only those roles defined in that Domain will be added to the Role list. The domain name is not case sensitive.

SecurityEnabled. Enables or disables security.

AuthenticationSource. Sets how the server determines the current user. "AuthNT" uses the operating system's security to get the Username.

LogonFailPage. The LogonFailPage attribute identifies a page where users are sent when an attempted logon fails. The page can either be your own logon page, or a page that gives the user additional instructions and will guide them to the main logon page for the application.

Note:
The element and attribute names are always case sensitive.

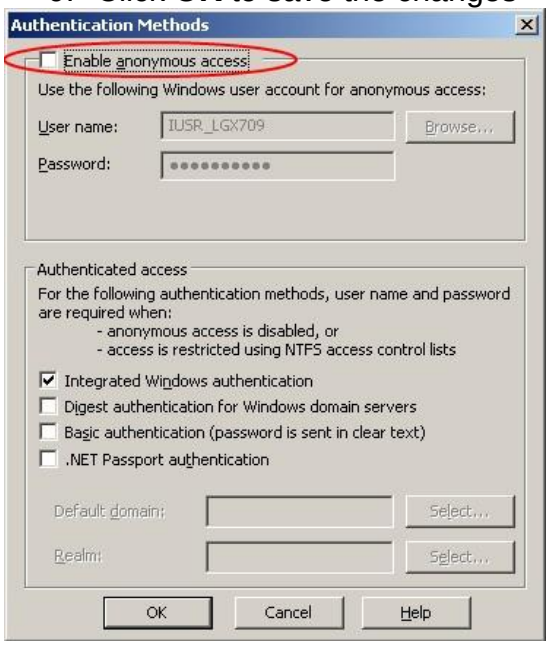
Note:
If the NTAuthenticationDomain attribute value is not specified, the user will be created as "domain\username" and the NT User Groups list will be "domain\NT User Group name". This means that the matching roles in Ad Hoc would also have to prepend the domain in the role name.

Disable anonymous access for the Ad Hoc virtual directory

A virtual directory (pre-IIS 7.0) or Web application (IIS 7.0) for Ad Hoc is created and configured during the initial creation of the Ad Hoc instance. The Management Console configures the virtual directory for *anonymous* access to support the built-in security features. Administrators must disable anonymous access to use NT authentication. In addition, **Integrated Windows authentication** must be enabled.

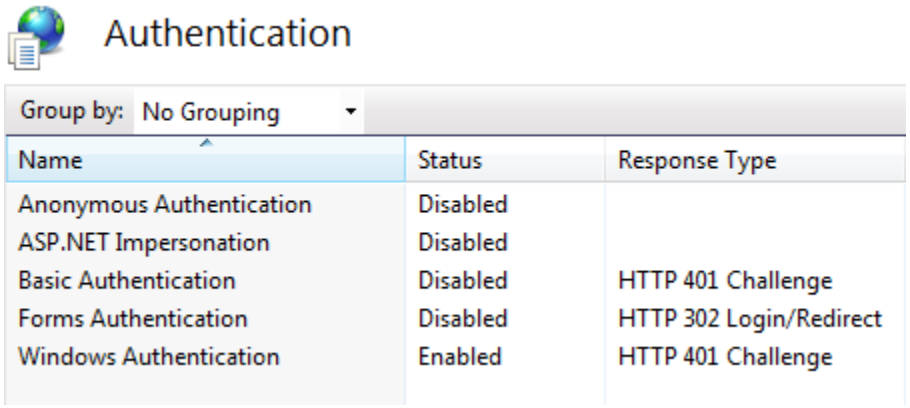
To disable anonymous access to Ad Hoc in IIS pre-7.0:

1. Launch IIS
2. Locate the virtual directory specified during the creation of the Ad Hoc instance
3. Right-click the virtual directory and choose **Properties**.
4. Click the **Directory Security** tab
5. Click **Edit** to modify the Authentication and access control method.
6. Uncheck the **Enable anonymous access** checkbox
7. Verify that the **Integrated Windows authentication** is checked
8. Click **OK** to dismiss the dialog
9. Click **OK** to save the changes



To disable anonymous access to Ad Hoc in IIS 7.0:

1. Launch IIS
2. Click on the Web Application specified during the creation of the Ad Hoc instance
3. In the *IIS* panel, double-click on the **Authentication** icon
4. Right-click on **Anonymous Authentication** and disable this attribute
5. Right-click on **Windows Authentication** and enable this attribute
6. Close IIS



Set the *ahUserGroupID* session variable

Ad Hoc must be able to determine the organization that a user is a member of. For new users, the organization ID must be passed into Ad Hoc as part of the process of accessing the instance. Typically in an NT Authentication scenario all users belong to the same organization; however that is not a requirement.

The session variable that identifies the organization is the *ahUserGroupID* (case sensitive). This session variable must be set before control is passed to the Gateway.aspx page. Also, session variables cannot be passed between web applications. They must be created within the application where they are intended to be used.

One technique to set this session variable value is to code it into the *Default.aspx* file. The *Default.aspx* file is found in the root folder specified during the creation of an Ad Hoc instance. To set the ahUserGroupID through the *Default.aspx* file:

1. Make a backup of the *Default.aspx* file
2. Edit the *Default.aspx* file
3. Insert the following XML: <% Session("ahUserGroupID") = 1 %> just before the <html> element to identify the organization associated with the user
4. Save the file

The *ahUserGroupID* session variable may also be passed from a parent application, included as a parameter in the URL, or set from a login script.

In most NT Authentication scenarios, the normal login page is bypassed. Extending the technique described above, the *Default.aspx* file could be modified to launch Ad Hoc directly.

Following is an example of a *Default.aspx* page that sets the organization session variable and launches the Ad Hoc application.

```
<%@ Page Language="VB" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<% Session("ahUserGroupID") = 1 %>
<% Response.Redirect("Gateway.aspx") %>
```

Notes:

1. *On any error on login the user is redirected to the page specified in the LogonFailPage attribute of the <Security> element. An aspx page called LoginError.aspx has been included in the installation package which can be set as the LogonFailPage.*
2. *For NT Authentication, the LogonFailPage attribute should **not** be set to Default.aspx if Default.aspx has a Response.Redirect in it. In this case login errors may cause an infinite loop. Using the provided LoginError.aspx page or a custom designed page may avoid this. The LoginError.aspx page simply displays the error message thrown by Gateway.aspx from a session variable called "rdLogonFailMessage".*