Directory Manager

DIRECTORY MANAGER V1.6
Quick Start Guide

Directory Manager is an easy-to-use, customizable, Web-based utility that allows the administrator to delegate the ability to update user's information in the Active Directory to non-admin users such as the HR department or a receptionist.

This document is intended to provide you with a quick reference for getting started Directory Manager.

Tips

Here are some tips and information that will make your work with Directory Manager easier and more trouble-free.

- All updates to the Active Directory are done using a service / proxy account, not via the currently logged on user's credentials.
  - Create a service/proxy account that has the permissions necessary to update all users.
  - The service/proxy account's password should not expire.
  - Members of Account Operators cannot update other operator or administrative users. This is a Windows security feature.
- Users are given permission to use Directory Manager by creating a **Directory Update Managers** group in Active Directory and putting the end users in this group. No permissions need to be assigned to this group.
- Almost all customization and configuration is performed in one of three XML files found in the c:\inetpub\wwwroot\directory
  - DirectorySettings.XML contains the field/attribute configuration such as which fields show up on the Edit interface, which fields are hidden, drop-down list values, etc…
  - AppSettings.XML allows you to customize the search options, window labels, error/help messages, and narrow the scope of a search to a single OU.
  - AddressSettings.XML allows you to specify an attribute such as Office name that, when selected, will automatically read the office address (street, city, state, postal/zip code, and country) and populate that for the user.
- Get yourself a good XML editor; that will make editing the XML files much more painless. We recommend Notepad++; a very good and free text editor. http://notepad-plus.sourceforge.net
- Always make backup copies of your XML files.

- Photo support requires that the NETWORK SERVICE be given "Modify" permissions to the c:\inetpub\wwwroot\directorymanager\photos folder.
- You can install the evaluation version and later run the Configuration utility to add a license key so that it no longer expires.

Prerequisites

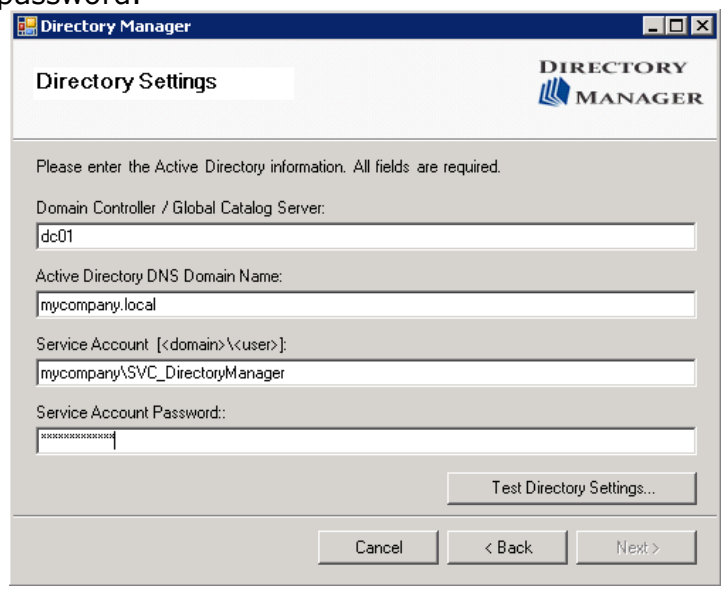
Make sure that your server meets all of the prerequisites and your installation will go much more smoothly.

- Server requirements
  - Windows Server 2003 with SP1 or later
  - IIS Web service installed
  - Microsoft .NET Framework v2.0 and v3.5 installed
  - Server must be a member of the domain/forest
  - In IIS Manager under Web Service Extensions, make sure that ASP.NET v2.0.50272 is visible and Allowed
- Windows Server 2008 requirements
  - Windows Server 2008 / Windows Server 2008 R2
  - IIS 7 / IIS 7.5 web service enabled
  - IIS 6 compatibility components of IIS 7 enabled
  - ASP.NET enabled
  - .NET Framework 3.5
  - Apply all Microsoft critical and recommended updates
- Create a service/proxy account that has permissions to update user accounts. All updates to the Active Directory are performed using this account, NOT the end user's account.
- Have a domain user account that is also a member of the IIS Server's local Administrators group to do the installation
- If User Access Control (UAC) is enabled, you may need to run the installer from the command prompt, such as this:
  - msiexec.exe /i c:\directorymanager.msi
- Create a **Directory Update Managers** group that will hold the authorized Directory Manager users. Directory Manager users need no special permissions in Active Directory other than membership in this group.
- Download the latest version of the software from our Web site.

Installation

Installation is usually simple and quick though the software will be installed with the default XML templates and you will need to customize these for your organization.

1. Run the **DirectoryManager.MSI** installer
2. You can take most of the defaults for the installer including the default virtual directory name (**/DirectoryManager**) and putting the site on to the Default Web Site.
3. On the Directory Settings screen, enter the domain controller name, the domain name, the service account information, and the service account password.



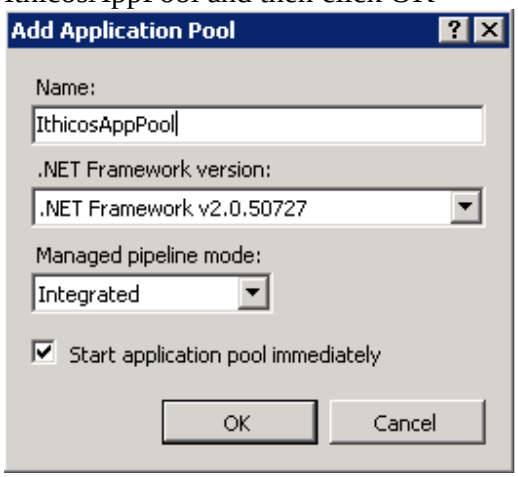
4. Click the Test Directory Settings button and click Next
5. Enter the organization name and the license key (or check evaluation version)
6. Finish the installation
7. If on a domain controller, you may need give the NETWORK SERVICE account "Modify" permissions to the folder:
   `C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\Temporary ASP.NET Files`
8. If using the Photo feature, give the NETWORK SERVICE account "Modify" permissions to the folder:
   `c:\inetpub\wwwroot\DirectoryManager\Photos`
9. Customize the XML files to suit your organization.

Creating an Application Pool

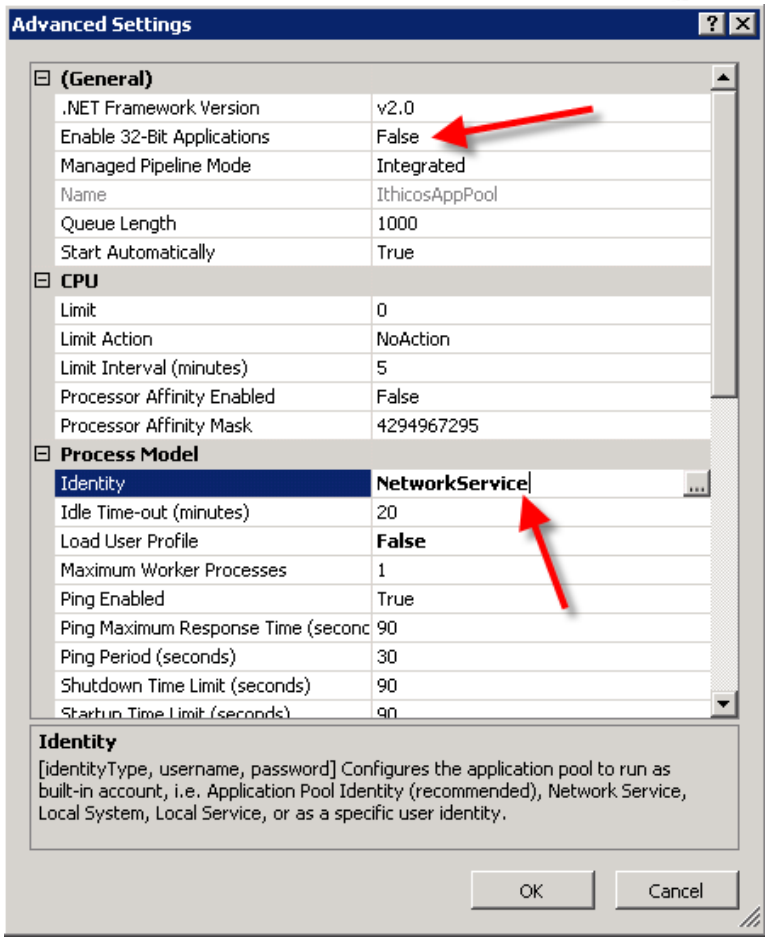
We strongly recommend that you create a dedicated application pool for Ithicos software. An IIS application pool is essentially a dedicated memory space and processor threads for

running a Web application. Creating a dedicated application pool allows our applications to run in their own memory space, not interfere with other Web applications, prevent other Web applications from interfering with us, and to run in a specific security context. To create an application pool called IthicosAppPool on a W2K8 server, follow these steps:

1. On your web server, open Internet Information Services Manager
2. Navigate to the Application Pools container
3. Right Click on Application Pools and choose “Add Application Pool”
4. In the Add Application Pool dialog box, enter the application pool name, such as IthicosAppPool and then click OK



5. Locate the newly created application pool in the Application Pools container, right click on it and choose Advanced Settings

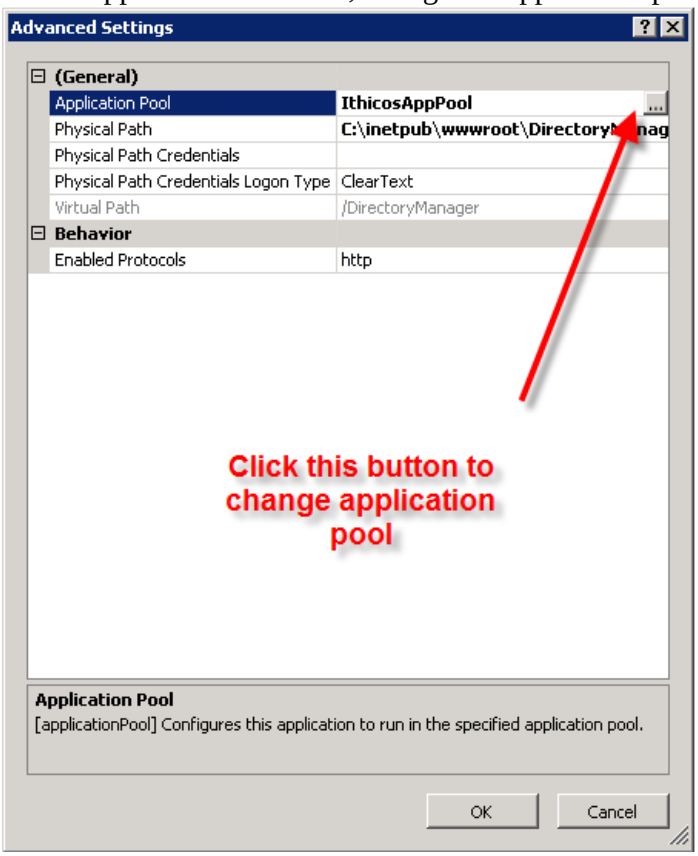


6. Ensure that in the Advanced Settings dialog box that the Enable 32-Bit Applications is set to “False” and that the Identity is changes so that it is “NetworkService”.
7. Click OK to save your changes

The application pool you create can be used for all Ithicos Solutions products. It is easy to change a Web application to another application pool. Here is an example for Directory Manager.

1. Open Internet Information Services Manager and navigate to the Web site that has the DirectoryManager virtual directory / application.
2. Right click on DirectoryManager and choose Manage Application -> Advanced Settings

3. In the Application Pool box, change the application pool to be IthicosAppPool



4. Click OK to save your changes
5. You may have to run IISRESET from the command prompt for the changes to take effect.

Using Secure Sockets Layer (SSL)

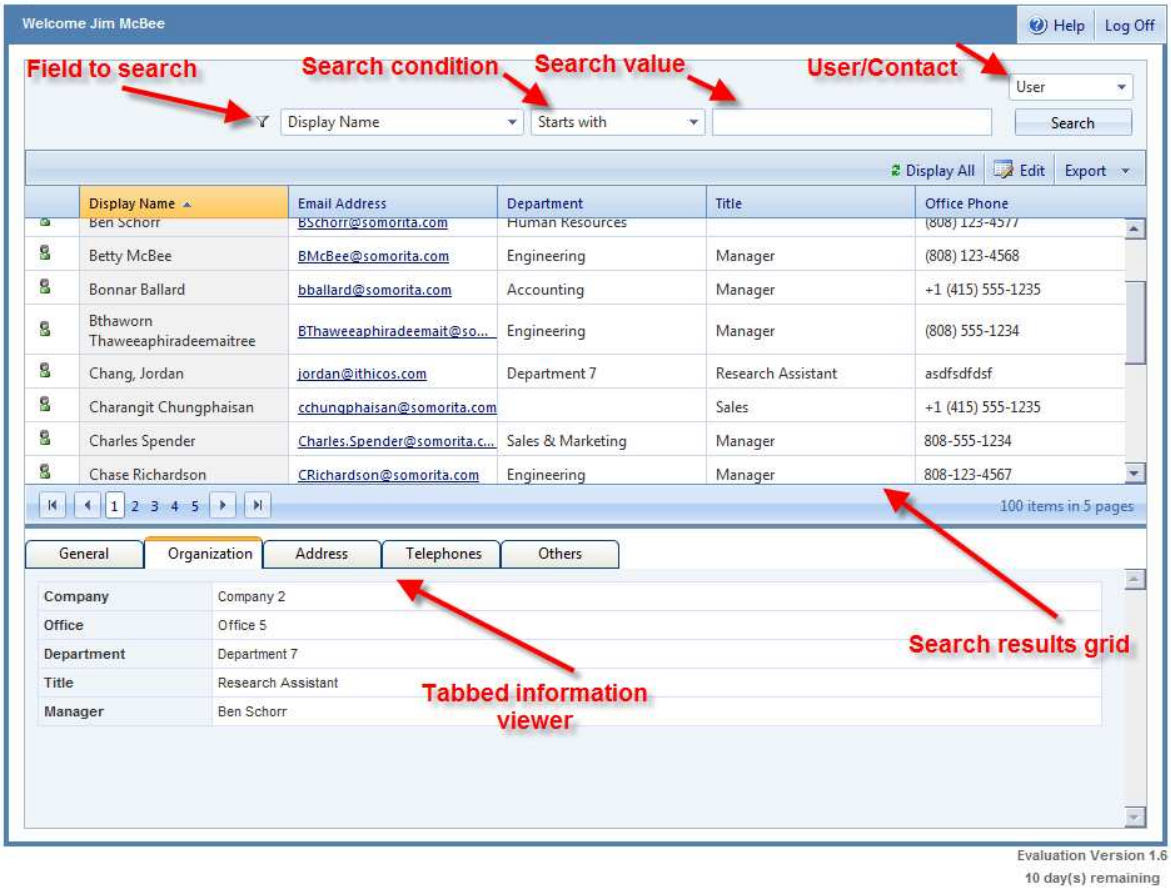
We recommend that you implement Secure Sockets Layer (SSL) for any web site on which end users will enter private/personal data or on which a username / password may be passed over the network. All web-based Ithicos Solutions products will work on SSL-enabled web sites.

There is nothing you need to do to our products to enable SSL. This is done in Internet Information Server (IIS) 6, IIS 7, or IIS 7.5. For more information, see:
http://support.microsoft.com/kb/299875
http://learn.iis.net/page.aspx/144/how-to-set-up-ssl-on-iis-7/

We recommend using a certificate authority that will be trusted by the browsers of all users. It is a very bad practice to get users in the habit of ignoring SSL security warnings.

Directory Manager Interface Overview

Directory Manager's user interface consists of two distinct components. The first is the search screen:



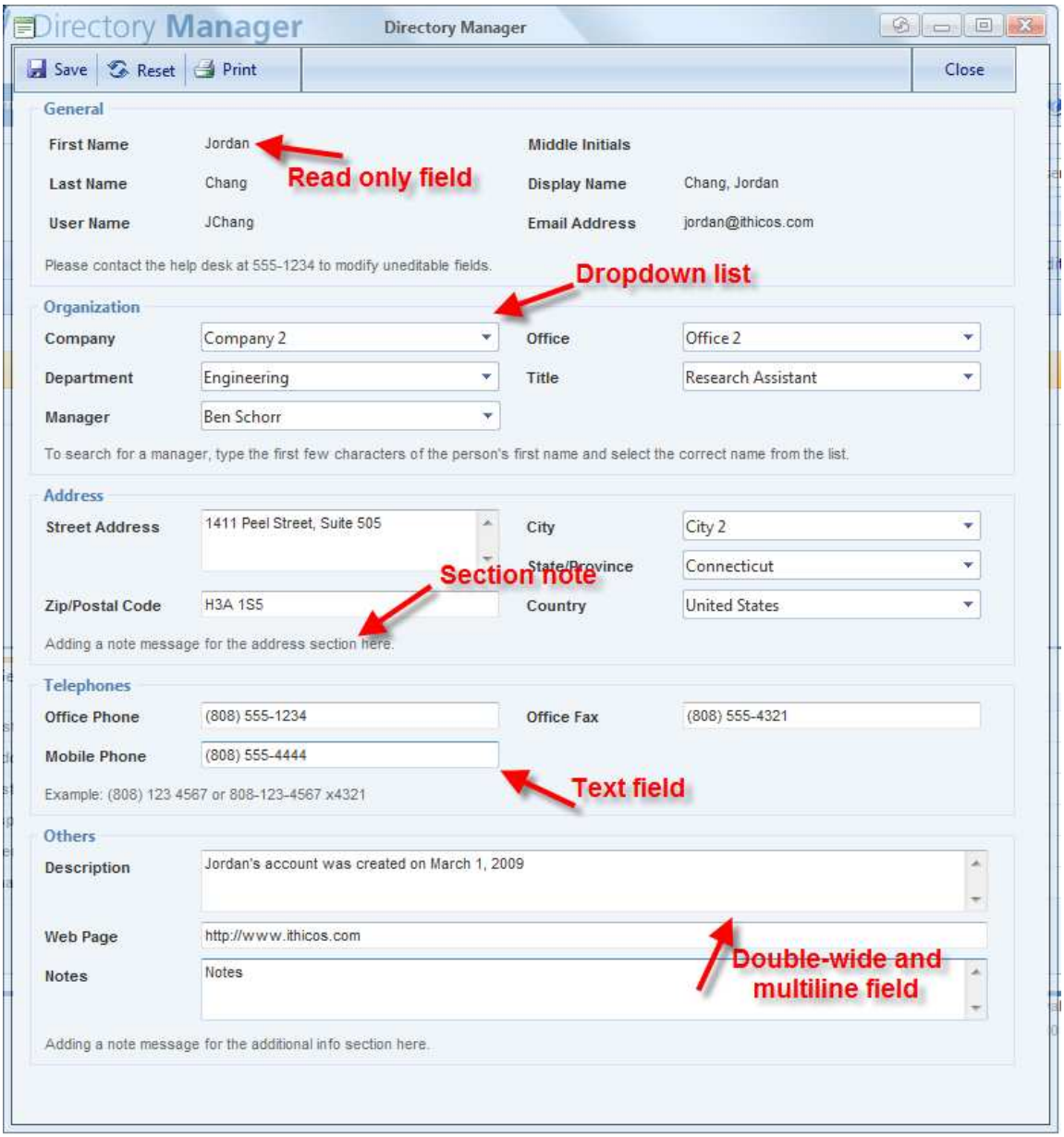
While there are many configurable elements on the search screen, the ones of importance include:

- The search options which includes the fields available to search, the search conditions (Starts With, Ends With, Equals…), and the actual value to search.
- The user/contact drop-down box displays the types of objects to show in the search list: users, contacts, or both
- The search results grid including the columns configured to display in the search grid.

- The tabbed viewer that displays properties of the currently selected object in the search grid. Note that the tabbed viewer does NOT permit editing of the selected object. You must double-click on the object or select the object and click Edit.

Note that most of the search screen is controlled and customized through the AppSettings.XML file.

The second part of the user interface that is noteworthy is the edit screen. The edit screen is mostly customized in the DirectorySettings.XML file.



Directory Manager

Customization Quick Reference

Customizing the Directory Manager interface for your specific requirements is reasonably simple once you have taken a look at the XML files that store the configuration and if you get yourself a good XML editor. The XML files are all found in the following folder:
`C:\inetpub\wwwroot\DirectoryManager\Settings`

There are four primary configuration files available to you for Directory Manager:

DirectorySettings.XML – Controls the user interface, which fields are visible, which fields are editable, field labels, validation rules, dropdown lists, etc…

AddressSettings.XML – Allows a user to populate multiple values, such as street address, city, state, country, postal code, by selecting a single value such as an office name.

Subsettings.XML – Allows the creation of a parent-child relationship between two fields, such as division to department. If a user selects a certain division name, then only the departments associated with that particular division are selected.

AppSettings.XML – Controls settings in the user interface such as button labels, search fields, export fields, and which fields are visible in the search results columns.

Each attribute or field on the user interface has some of the same basic options:

- Each field label can be changed
- Each field type can be text, dropdown, or combo
- Each field can be hidden / unhidden
- Each field can be made read only or editable
- Each field can have a default value
- Each field can have a validation format assigned to it
- Each field can have example text

Below is an example of the Organization section and two fields (company and office.) This will give you an idea of some of the types of customization that can be done.

```
44    <!-- ORGANIZATION SECTION -->
45    <organization label="Organization" visible="yes">
46      <company label="Company" type="dropdown" visible="no" editable="yes" required="no"
validationFormat="" defaultValue="" example="">
47        <value>Ithicos Solutions</value>
48        <value>Somorita Surfboards</value>
49        <value>Volcano Surfboards &amp;amp; Smoothies Shop</value>
50        <value>Bob's Boogie Boards</value>
51      </company>
52      <office label="Office" type="text" visible="no" editable="yes" required="no" validationFormat=""
defaultValue="" example=""></office>
```

The DirectorySettings.XML and AddressSettings.XML files CAN be interchanged with **Directory Update**. The AppSettings.XML file cannot be used with our other programs.

The following are the attribute/field options and possible values:

Option	Description and possible values
label	Allows you to change the text label on the interface
type	Sets the field type. Options are “text”, “dropdown”, and “combo”
visible	Sets the attribute to visible or hidden. Options are “yes” or “no”
editable	Allows the user to either edit the field or have it appear read-only. Options are “yes” or “no”.
required	Sets the field so that a value must be entered or selected. Options are “yes” or “no”.
validationFormat	Allows you to specify a validation format name (defined at the bottom of the XML file) that controls the required format/structure.
defaultValue	Allows you to specify a default value for the field. If using a dropdown list, the default value must be in the dropdown list also. The default value will only be used if the Active Directory attribute is empty.
example	Allows you to specify sample text that will show directly below the field. As long as it is set to example=””, the sample text will not show up.

Here is the company name tag that has been configured as a drop-down list. Notice that for company, within the "open" and "close" tags there are tags for each option in the drop-down list.

```
<company label="Company" type="dropdown" visible="no" editable="yes"
required="no" validationFormat="" defaultValue="" example="">
      <value>Ithicos Solutions</value>
      <value>Somorita Surfboards</value>
      <value>Volcano Surfboards &amp;amp; Smoothies Shop</value>
      <value>Bob's Boogie Boards</value>
</company>
```

Authorized Users of Directory Manager

By default, members of the domain's Account Operators, Administrators, and Domain Admins groups can login to Directory Manager and update users. The intention of Directory Manager is to allow non-administrators to be able to edit users (or contacts) in the Active Directory without giving them the keys to the kingdom.

In Active Directory, create a group called **Directory Update Managers** and put the users that should be authorized to use Directory Manager in that group. They need no additional permissions. Actual updates to the Active Directory are performed using the security credentials of the service/proxy account, not the user's account. Membership in the **Directory Update Managers** group merely authorizes the user to use Directory Manager.

Customizing the Search Options and Search Results Grid

The attributes on which we allow searching and the columns shown in the search results grid are the defaults that we have found work best for most of our customers but they may not be ideal for your organization. This can be changed via the AppSettings.XML file. Locate the **userList** section of that file; within that section is a tag called **<columns>**. Under the <columns> section, there is a tag for each attribute on the interface. This section is shown in the screen capture below.

```
    <userList maxResults="100" pageSize="20" sortBy="displayName" showOnlyExchangeEnabledUsers="no"
showDisabledUsers="no" showInitialResults="yes" showDetailPanel="yes">
      <columns>
        <personalTitle          headerText="Personal Title"     visible="no" filter="no" export="no" />
        <firstName              headerText="First Name"         visible="no" filter="no" export="no" />
        <initials               headerText="Middle Initials"    visible="no" filter="no" export="no" />
        <middleName             headerText="Middle Name"        visible="no" filter="no" export="no" />
        <lastName               headerText="Last Name"          visible="no" filter="no" export="no" />
        <nameSuffix             headerText="Name Suffix"        visible="no" filter="no" export="no" />
        <displayName            headerText="Display Name"       visible="yes" filter="yes" export="yes" />
        <email                  headerText="Email Address"      visible="yes" filter="yes" export="yes" />
        <userName               headerText="User Name"          visible="no" filter="yes" export="no" />
        <company                headerText="Company"            visible="no" filter="no" export="no" />
        <office                 headerText="Office"             visible="no" filter="no" export="no" />
        <division               headerText="Division"           visible="no" filter="no" export="no" />
        <department             headerText="Department"         visible="yes" filter="yes" export="yes" />
        <departmentNumber       headerText="Department #"       visible="no" filter="no" export="no" />
        <title                  headerText="Title"              visible="yes" filter="yes" export="yes" />
        <employeeId             headerText="Employee ID"        visible="no" filter="no" export="no" />
        <employeeNumber         headerText="Employee #"         visible="no" filter="no" export="no" />
        <employeeType           headerText="Employee Type"      visible="no" filter="no" export="no" />
        <manager                headerText="Manager"            visible="no" filter="yes" export="no" />
        <assistant              headerText="Assistant"          visible="no" filter="no" export="no" />
        <secretary              headerText="Secretary"          visible="no" filter="no" export="no" />
        <officePhone            headerText="Office Phone"       visible="yes" filter="yes" export="yes" />
        <otherOfficePhone       headerText="Other Office Phone" visible="no" filter="no" export="no" />
        <fax                    headerText="Office Fax"         visible="no" filter="no" export="no" />
```

Let's take as an example the department field/attribute:

```
<department headerText="Department" visible="yes"
filter="yes" export="yes" />
```

- The headerText option sets the label for the column.
- Visible="yes" shows the field in the columns listing
- Filter="yes" allows to you search by department name
- Export ="yes" allows this attribute to be exported to a Excel spreadsheet or text file.

Maximum Users Returned

You may have noticed that the default search as well as any other search you do in Directory Manager returns a maximum of 100 search results. This is by design so that we don't overwhelm a domain controller with too many LDAP search requests. This filtering feature as well as several other filter features is configurable in the **userList** tag found in the AppSettings.XML file. Simply change the **maxResults** option found in the **userList** tag.

```
<userList maxResults="100" pageSize="20"
sortBy="displayName" showOnlyExchangeEnabledUsers="no"
showDisabledUsers="no" showInitialResults="yes"
showDetailPanel="yes">
```

The **userList** tag also allows you to filter out users that are not "mail-enabled" (if you use Exchange, and to filter out disabled users.

The Directory Manager interface was designed to "search" for users more than it was designed to "browse" through hundreds or thousands of user accounts. This is partially because of the protocol we use to query Active Directory (LDAP). If you want to display more than 1,000 users in a single search, you will need to also update the maximum search result size that Active Directory will return to an LDAP client. This is done by using the NTDSUTIL.EXE command. See this link for more information:
http://support.microsoft.com/kb/315071

Logging and Auditing

Directory Manager offers two forms of logging. The first is to log the last date/time of each update to an attribute in Active Directory. The second is to log each individual change to a tab-separated value (TSV) file. These are enabled in the auditing section of the AppSettings.XML file in the section shown here in Figure 1.

```
<!-- Auditing - Directory Update will log the date in yyyy-mm-ddThh:mm:ss format format, username, and IP address each time
<!-- For attribute audit, you can specify extensionAttribute1 through extensionAttribute15. In order to use the extensionAt
<!-- your Active Directory must have been prepped for Exchange 2000/2003/2007. -->
<!-- For log file audit, the default log folder is the "Logs" folder in the DirectoryUpdate folder.  You can change this to
<!-- Ensure that the "Application Pool" account (usually NETWORK SERVICE) has "Modify" permissions to the Logs folder. -->
<auditing>
  <auditingAttribute enabled="no" attribute="extensionAttribute11" showUserLastUpdate="yes" text="Your last update was" />
  <auditingLogFile   enabled="no" logFileFolder="c:\inetpub\wwwroot\directoryupdate\logs\">
    <headers>
      <dateTime  text="Date/Time" />
      <userName  text="User Name" />
      <sourceIp  text="Source IP" />
      <fieldName text="Field Name" />
      <oldValue  text="Old Value" />
      <newValue  text="New Value" />
    </headers>
  </auditingLogFile>
</auditing>
```

Figure 1: Auditing section of AppSettings.XML file

Note that the default log file folder is **c:\inetpub\wwwroot\directoryupdate\logs**. The user account under which the application pool is running (usually NETWORK SERVICE) must have Modify permissions to this folder.

Integrated Windows Authentication / Single Sign On

Internet Information Server (IIS), Windows, and Internet Explorer support a feature called Integrated Windows Authentication.

Essentially this means that when a user connects to a web site that supports this feature (and if the Web browser supports it), then the user will automatically be signed on.

The default logon interface for Directory Manager is the logon form also known as Forms Based Authentication. Forms Based Authentication (FBA) should work with any browser, from any Windows, Unix, or Apple desktop computer and for users that are not on a computer that is a member of a domain.

You can use Integrated Windows Authentication (IWA) with Directory Manager to eliminate the logon for the user. All the user needs to do is to visit the URL for Directory Manager and they are automatically logged on. Note that there is no such thing as a "log off" when using IWA since the user did not really logon.

To enable IWA for Directory Manager, edit the **web.config** file (found in **c:\inetpub\wwwroot\directorymanager** by default). Locate the authentication section and change *mode="Forms"* to *mode="Windows"*

```
    <customErrors mode="RemoteOnly"/>
    <!--  AUTHENTICATION
      This section sets the authentication policies of the application. Possible modes are "Windows",
      and "Forms".

      "Windows" IIS performs authentication (Basic, Digest, or Integrated Windows) according to
       its settings for the application.
      "Forms" You provide a custom form (Web page) for users to enter their credentials, and then
       you authenticate them in your application. A user credential token is stored in a cookie.
-->
    <!--
       The <authentication> section enables configuration
       of the security authentication mode used by
       ASP.NET to identify an incoming user.
    -->
    <authentication mode="Forms">
       <forms name="AppNameAuth" path="/" loginUrl="Login.aspx" protection="All" timeout="60"/>
    </authentication>
```

Integrated Windows Authentication works provided the following is true:

- The user is using a browser that supports IWA such as Internet Explorer
- The computer on which the user is logged in is a member of the Active Directory forest in which the Directory Update IIS Server is located
- The user logs on to that computer with a domain account
- There are no security settings that prevent IWA
- The browser's local security zone permits IWA (such as Internet Explorer's "Local Intranet" zone.

If you are interested in learning more about IWA, see this link:
http://en.wikipedia.org/wiki/Integrated_Windows_Authentication