Directory Update v2.x
Release Notes

Ithicos Solutions is pleased to announce the release of Directory Update v2.0; this release includes many features that our customers have requested over since the release of Directory Update v1.7. This release also includes some "nice to have" features that we have dreamed up on our own.

New Features

The following are new features and changes included in Directory Update v2.0:

- Change Password feature that allows an end user to change their own password if they know their current password
- Integration with Directory Password v1.0 (an additional cost product) that allows a user to reset their password if they forgot their password or to unlock their account if it is locked.
- Phone number format can be based on country code selected
- Subsets feature that allows parent-child relationship between fields (via the subsettings.XML file)
- E-mail notifications upon change to user, user's manager, and/or predefined SMTP address
- Photo attribute default changed to use *thumbnailPhoto* and size of 128x128 – this better aligns with Microsoft's future plans.
- Ability to log change / auditing to a tab separated value (TSV) text file
- Updates and fixes to support Internet Explorer 8.x and Firefox 3.x (and later)
- First name, middle initial, last name, and user name are no longer visible on the interface by default.
- Improved error handling and friendlier error messages
- Internet Explorer 8.x and 9.x as well as Firefox 4.x and later are supported. Other browsers *may* work but we do not test with them. Note that Internet Explorer 7 compatibility mode is not supported and should not be used.
- The .NET Framework v3.5 is required

Notepad ++

If you are still editing your XML files using a boring editor like Notepad, we strongly urge you to download the free Notepad ++ editor. It will make editing our XML files much easier because it puts everything in color and it makes common errors such as not closing a tag much easier to find. You can find Notepad++ at http://notepad-plus-plus.org/.

License Keys

Directory Update v2.0 requires different license key from previous versions. If you are eligible for support and upgrades, please contact support @ ithicos.com. Include your organization name and the DNS domain name of your Active Directory domain.

If your maintenance agreement has expired, please see the Purchasing section of our Web site (http://www.ithicos.com) for information on how to active/extend your maintenance period.

Windows Server Requirements

Directory Update v2.0 runs on the following operating systems:
- Windows Server 2003 SP2 (x86 or x86) – Standard or Enterprise
- Windows Server 2003 R2 with SP2 (x86 or x64) – Standard or Enterprise
- Windows Server 2008 SP2 (x86 or x64) – Standard or Enterprise
- Windows Server 2008 R2 – Standard or Enterprise

The following are additional requirements:
- IIS 6 or IIS 7
  - If IIS 7 is used, the IIS 6 compatibility components must be enabled
  - If IIS 6 is used, ASP.NET v2.0 must ben enabled in Web Extensions
- ASP.NET must be enabled
- The .NET Framework v3.5 is required

We strongly encourage you to apply all Microsoft critical and recommended updates using Microsoft Update.

Photo Support

Directory Update v2.0 and v2.1 use the **thumbnailPhoto** attribute to store photos. The default photo size is 128x128. We strongly recommend you keep these settings as this will allow the photo to properly display in Microsoft Outlook 2010 and the Microsoft Lync clients.

When a photo is uploaded to Active Directory, we will re-render the photo to 128x128 (a square aspect ratio). We recommend that photos be square to begin with so that the aspect ratio is not changed.

Most photos end up being around 7KB to 10KB one they are uploaded to Active Directory. So, if you have 1,000 users, this will increase the size of your Active Directory database by no more than 10MB.

When photos are uploaded to the Directory Update site, they are temporarily stored in the c:\inetpub\wwwroot\directoryupdate\photos folder. The IIS server must have

permissions to write files and delete files from this folder. By default, that should be the **Network Service** user.

Supported Browsers

We test against a specific set of Web browsers. As with many Web 2.0 applications, updates to these browsers may break an application. The following Web browser's are supported in Directory Update v2.0:

- Internet Explorer 8.x and 9.x
- Firefox 4.x and later

Other browsers such as Chrome or Safari may work. Internet Explorer 7 compatibility mode is not supported and should not be used.

Upgrading from Previous Versions

We recommend using the *new* XML files that come with Directory Update v2.0 to ensure that all new features are available. This means that you have re-create or copy-and-paste your data from your old XML files.

Directory Update v2.0 *can* use the same DirectorySettings.XML and AddressSettings.XML files as v1.6; this can simplify upgrading from v1.6. However, if you have customized other features such as the AppSettings.XML file, the style.css file, or other files, then you will need to make those changes again. Here is the upgrade path we recommend:

1. Make a backup copy of your c:\inetpub\wwwroot\directoryupdate folder
2. In Control Panel -> Add/Remove Programs, remove Directory Update
3. Delete the c:\inetpub\wwwroot\directoryupdate folder
4. Install the new version of Directory Update
5. Backup the new c:\inetpub\wwwroot\directoryupdate\settings\DirectorySettings.xml (and the new AddressSettings.XML) file
6. Replace the new DirectorySettings.xml file with the one from your previous installation. If you customized the AddressSettings.XML file, then replace the new AddressSettings.XML file with your original file from the previous installation.
7. Customize the new AppSettings.XML and other files if necessary.

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
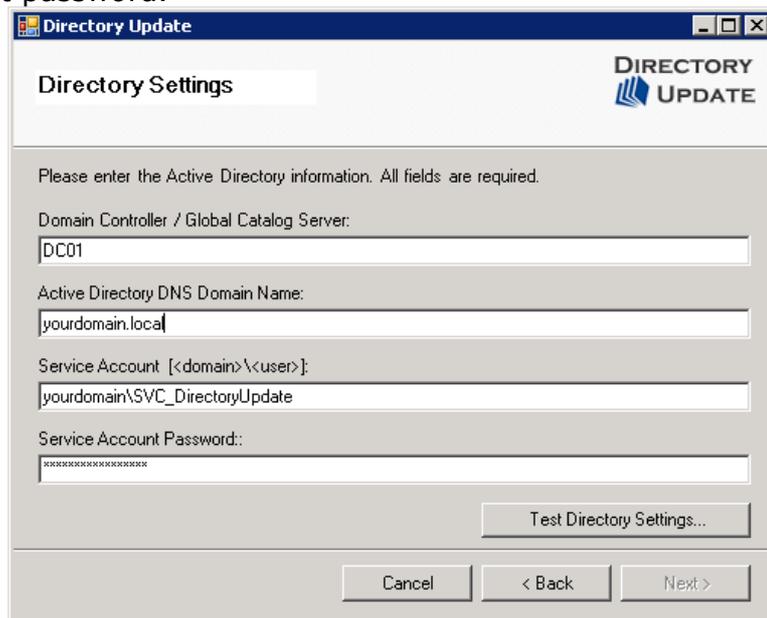http://learn.iis.net/page.aspx/144/how-to-set-up-ssl-on-iis-7/

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.

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 **DirectoryUpdate.MSI** installer
2. You can take most of the defaults for the installer including the default virtual directory name (**/DirectoryUpdate**) 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, 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\DirectoryUpdate\Photos`

9. Customize the XML files to suit your organization.

Note when installing on Windows Server 2008 that if User Access Control (UAC) is enabled, you may get a permissions error even if you are logged on as a domain administrator. Open up a command prompt as an administrator and run this command:

msiexec.exe /i c:\directoryupdate.msi

This can bypass the permissions error.

Upgrading from Versions Older than v1.6

The XML files from v1.5 and earlier is NOT compatible with v1.7. This means you have to re-create all XML files using the new XML files as a base. Essentially, the upgrade process is to backup your old installation to another folder, remove Directory Update, install the new version, and then customize the NEW XML files. Versions prior to v1.6 had most all configuration including field definitions and drop-down list values in the AppSettings.XML file.

Once you install the new version, you will need to make your custom configuration changes to the new XML files and Style.css file if applicable. You can take your drop-down list values from your previous versions and copy them to the new DirectorySettings.xml file, though. Here are the steps we recommend to upgrading:

1. Make a backup copy of your c:\inetpub\wwwroot\directoryupdate folder (or where ever you installed the software originally.)
2. Run Control Panel -> Add/Remove Programs and remove the previous version of Directory Update
3. Delete the HKLM\Software\ITCS Hawaii or the HKLM\Software\Ithicos Registry key
4. Delete the directory c:\inetpub\wwwroot\directoryupdate (or where ever you installed Directory Update originally.)
5. Install the new version of Directory Update (using the DirectoryUpdate.msi installer)
6. Copy your customized <value> </value> tags from the original AppSettings.xml file and add them to the new .\Settings\DirectorySettings.xml file
7. Make additional customizations to the AppSettings.xml and DirectorySettings.xml files such as changing labels, hiding fields or making fields required.

The trickiest part of the upgrade is moving your existing <value> tags to the new DirectorySettings.xml file. We recommend loading the old AppSettings.XML or DirectorySettings.XML file in to a text editor and then open the new DirectorySettings.XML file in another window; then you can copy and paste values between the editors.

IIS Application Pools

Simply put, an application pool is a separate memory space and process in which a web application can run. The "identity" of that application pool provides the security context in which the applications run. It is a good security practice to ensure that applications run with no more privileges than necessary.

We recommend that you create a separate application pool such as DirectoryUpdateAppPool. Application pools are created using the Internet Information Services (IIS) Manager console. In W2K3, simply copy the default application pool. In W2K8 (shown in Figure 1,) create a new application pool that uses the .NET Framework v2.0.50727.
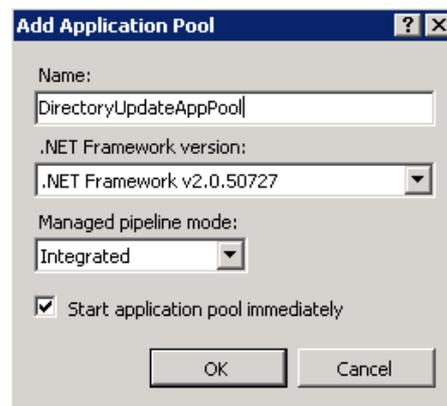

Figure 1: Creating an application pool in W2K8

If you are using Windows Server 2008, once you have created the new application pool, ensure that the identity is using the **NetworkService** user account:
1. In IIS Manager, under the application pools container, right click on the new application pool and choose Advanced Settings.
2. Locate and select the Identity option (under Process Model)
3. Click the small button to the right of the Identity box and select the built-in account NetworkService
4. Click OK twice to close the Advanced Settings.

Simplified DirectorySettings.XML file

In previous versions of Directory Update, we required that each field "tag" have all of the options available for that particular tag. Below is an example of the Company name tag:

```
    <company label="Company" type="dropdown" visible="yes"
editable="yes" required="no" validationFormat="" defaultValue="Ithicos
Solutions" example="">
```

Notice that the tag includes the label, field type, visible, editable, required, validation format, default value, and example options. These had to be inside the tag even if the option was not used for that particular tag. Directory Update v1.7 includes some new options such as multi-line and double-wide field tags. This would have made each tag even longer.

We have stripped back the options required for each tag to the bare minimum. Now, each tag has label, type, visible, and editable option.

```
<company label="Company" type="dropdown" visible="yes" editable="yes">
```

You can add additional options that you may require based on a specific field's requirements. Table 1 shows the options that are available and valid settings for those options.

Table 1: Valid options for the DirectorySettings.XML file

Tag option	Valid settings
label	Any text string that defines that particular field
type	"text" - Free form text field "dropdown" - Provides the user a drop-down list "combo" - Dropdown list that allows text entry "maskedtext" - Allows for the use of masked text validation
visible	"yes" "no"
editable	"yes" "no"
validationFormat	Any validation format that is defined in at the bottom of the DirectorySettings.XML file.
required	"yes" "no"
defaultValue	Any text string that should be the default for the field if it is empty.
example	Any text that should appear directly below the field that can be used as an example or a helpful tip.
mask	Define the mask for a maskedText field type such as "(###) ###-###" for a phone number field.
multiLine	"yes" "no"
doubleWide	"yes" "no"

Validating Phone Numbers Differently for Different Countries

Organizations that have offices in different countries may find a feature useful that allows you to use a different phone number format for each country. The first thing you must do is create a phone number format for each country that will be using and name the format the same as the country code. For example, the United States would be "US" while Australia would be "AU". You can find the 2 digit country codes in the DirectorySettings.XML file.

```
8       <!-- Format validation uses regular expressions. Each fiel
9       <!-- See http://www.regexlib.com for many sample Regular E:
0       <validations>
1         <validation format="US" formatExample="Example: (808) 12:
2         <validation format="UK" formatExample="Example: +44 (0) :
3         <validation format="AU" formatExample="Example: +61 2 12:
4         <validation format="Norway-Phone" formatExample="Example
5         <validation format="Ireland-Phone" formatExample="Example
6         <validation format="US-ZipCode" formatExample="Example: :
7       </validations>
```
Figure 2: Validations section of the DirectorySettings.XML file

Next, for each phone number that you want validated, in the telephone number tag, set the validation format to "country". Use the actual word "country", not the country name.

```
<!-- TELEPHONE SECTION -->
<telephones label="Telephones" visible="yes">
    <officePhone      label="Office Phone"   type="text" visible="yes" editable="yes" validationFormat="country"></officePhor
    <otherOfficePhone label="Office Phone 2" type="text" visible="no"  editable="yes" validationFormat="country"></otherOffic
    <fax              label="Office Fax"     type="text" visible="yes" editable="yes" validationFormat=""></fax>
    <otherFax         label="Office Fax 2"   type="text" visible="no"  editable="yes" validationFormat=""></otherFax>
```

Finally, in the country tag in the DirectorySettings.XML file, set the option setPhoneValidation="yes" option.

```
<!-- List is sorted as typed below. You can put your most common countries at the top of the list.
<!-- Default value for coutry must be set to a two-letter country code i.e. US, GB, or AF. -->
<country label="Country" type="dropdown" visible="yes" editable="yes" setPhoneValidation="yes">
    <value code="US" countryCode="840">United States</value>
    <value code="GB" countryCode="826">United Kingdom</value>
    <value code="AF" countryCode="004">Afghanistan</value>
    <value code="AS" countryCode="016">American Samoa</value>
    <value code="AR" countryCode="032">Argentina</value>
    <value code="AW" countryCode="533">Aruba</value>
    <value code="AU" countryCode="036">Australia</value>
    <value code="AT" countryCode="040">Austria</value>
    <value code="BS" countryCode="044">Bahamas</value>
    <value code="BH" countryCode="048">Bahrain</value>
    <value code="BB" countryCode="052">Barbados</value>
```

Telephone Numbers in an Address Set

In v1.5, we introduced the Address Sets feature; this means that you can populate the AddressSettings.XML file with the street, city, state, postal code, and country information and then when a user selects a field such as Office, that information will automatically be populated. The information is read from the AddressSetting.XML and automatically populated in the user interface for the user.

Here is an example of a user section from the AddressSettings.XML file. Notice that there are <fax>, <officePhone>, and <extensionAttribute1) attributes inside of the Address Set.

```
<addressSetting value="Office 1">
    <streetAddress>Street Address 1</streetAddress>
    <city>City 1</city>
    <stateOrProvince>Alaska</stateOrProvince>
    <zipOrPostalCode>11111</zipOrPostalCode>
    <country>US</country>
    <officePhone>office phone 1</officePhone>
    <fax>fax 1</fax>
    <extensionAttribute1>extAtt 1</extensionAttribute1>
</addressSetting>
```

Technically, any attribute from the DirectorySettings.XML file can be added to an Address Set. However, if you add the attribute to a set, you must populate it with data; otherwise it will clear the data in the field.

E-mail Notifications

Directory Update v2.0 introduces e-mail notifications as a new feature. All e-mail notification settings are configured in the AppSettings.XML file. The first thing that must be configured is the e-mail server; we recommend you use the fully qualified domain name of an e-mail server that will accept mail anonymously.

The e-mail server settings are found in the <emailSettings…> tag near the bottom of the AppSettings.XML file. An example is shown in Figure 3.

```
      <!-- SMTP server is used for e-mail notifications.  If e-mail a
   <emailSettings>
      <smtp server=mailserver.volcanosurfboards.com" port="25" />
   </emailSettings>

</appSettings>
```

Figure 3: Configuring the SMTP e-mail server

The SMTP e-mail server that you specify must accept mail from the Directory Update server. If you are sending to e-mail addresses outside of your organization, then that server must allow relay for the recipients to which you are sending.

There are three types of events for which you can send e-mail notifications. These are:
- Directory information updates (such as phone, address, title, etc…)
- End user changes their own password
- User updates/changes their security questions (if Directory Password is installed.)

Figure 4 shows the e-mail notification settings that are used if a user changes their own personal information in the Active Directory.

```
<emailNotification enabled="no">
  <sender address="support@ithicos.com" name="Ithicos Solution support" />
  <!-- List E-mail addresses seperated by semi-colon (;). -->
  <receipientTo  user="yes" manager="no" addresses="" />
  <receipientCc  user="no"  manager="yes"  addresses="" />
  <receipientBcc user="no"  manager="no"  addresses="help.desk@ithicos.com" />
  <subject>User Information Update Notification</subject>
  <messageBody>
    <greeting>Dear</greeting>
    <message>
      You have sucessfully set up your information using the Directory Update appli
      If you are not the person who made the changes. Please notify the help desk.
    </message>
    <closing>Thank you</closing>
  </messageBody>
</emailNotification>
```
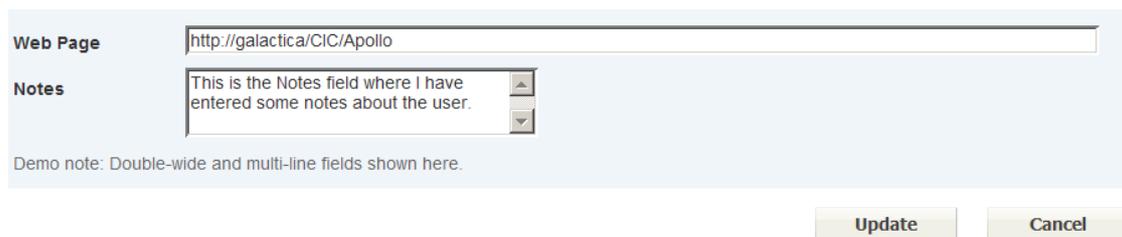
Figure 4: E-mail notification settings

You must customize the sender's SMTP address and display name since the default/example values will not be valid for your company. You can send the notification to the user, the user's manager, or a predefined SMTP address (such as the help desk's SMTP address.

Note that if you select the option to send the notification to the person's manager, the person's Manager field must be filled out and the manager must have an e-mail address.

Multi-line fields

A very common request for v1.6 customers was the ability to make some fields multi-line. The most common of these are the fields such as description and the notes field that can hold more text. You can make a field such as the telephone number field multi-line, but it probably won't do you much good.

Figure 5 shows an example of the Notes field as a multi-line field; if you define a field as multi-line the field will have three lines. The number of lines is not configurable.

Figure 5: Multi-line and double-wide fields

To make a field multi-line, simply edit the DirectorySettings.XML file and add the multiLine="yes" option to the field you want. Below is an example of how to make the Notes field a multi-line field.

```
<notes label="Notes" type="text" visible="yes" editable="yes"
multiLine="yes"></notes>
```

Double-wide fields

Another common request in all versions has been the ability to make a field wider than the standard width (such as the Notes field shown previously in Figure 5). We have a new option in v1.7 that allows you to configure a field to be 'double-wide'; this is also shown in Figure 5.

To make a field double-wide, you need to add a the doubleWide="yes" option to the tag you are interested in modifying. Below is an example:

```
<webPage label="Web Page" type="text" visible="yes" editable="yes"
doubleWide="yes"></webPage>
```

Masked Text Formatting

We are introducing a feature that allows you to control the way that text is entered in to a field by providing "input guidance" or field formatting such as the parenthesis or hyphens in a phone number. For example, using the masked text feature, a blank phone number field might look like this:

(_ _ _) _ _ _ - _ _ _ _

The user would have to enter data in the specified format. Any special characters, such as the parenthesis or the hyphen would ALSO be stored in the Active Directory.

The masked text option provides an alternative to using a Regular Expression (REGEX) to validate data and can be used in any type of field. However, the masked text option does not provide the same flexibility or level of validation that a REGEX does. You cannot use a REGEX and the masked text option.

There are things that have to be added to a field's tag in order to use that field with the masked text option. You need to set the field type to type="maskedText" and add a new option to the tag called mask which defines the mask format. Below is an example of the phone number tag BEFORE enabling the masked text feature:

```
<officePhone label="Office Phone" type="text" visible="yes"
editable="yes" validationFormat=""></officePhone>
```

Here is an example of the phone number tag using the masked text option. The phone number format is (808) 555-1234:

```
<officePhone label="Office Phone" type="maskedText" mask="(###) ###-
####" visible="yes" editable="yes" validationFormat=""></officePhone>
```

The mask option controls the format: `mask="(###) ###-####"` You can enter the parenthesis characters as well spaces and hyphens. The "#" character indicates the places where there must be a numeral.

Previous Release Notes

The below section is information taken from previous release notes.

Photo URL

In Directory Update v1.6, we introduced the ability to upload a photo to the Active Directory; the photo is stored in the *jpegPhoto* or the *thumbnailPhoto* attribute of the user. This increases the size of the user's object in the Active Directory by between 5KB and 10KB (on average). The photo object shows up on the top of the Directory Update interface as shown in Figure 6.

Figure 6: Enabling the photo feature

Once the photo is stored in the Active Directory, other applications that can access that attribute can also use the photo.

Some customers have asked for the ability to store the photo as a URL path rather than actually uploading the photo file to an Active Directory attribute. In v1.7, we allow you to store the photo either by uploading the file in to the Active Directory (either the user's *jpegPhoto* or the *thumbnailPhoto* attribute) or by storing the URL path to the photo.

We recommend that you use either the *thumbnailPhoto* attribute rather than the URL path. This will ensure that your directory remains compatible with any future plans Microsoft may or may not have planned with respect to photo support.

Below is the default photo tag from the DirectorySettings.XML file. This tag tells Directory Update to upload the photo as an octet stream and store it in the *jpegPhoto* attribute. The *type="file"* option configures the file option and the *attribute="jpegPhoto"* option configures which attribute you can use.

```
<photo label="Add/Change Photo" type="file" visible="yes"
editable="yes" attribute="jpegPhoto" width="130" height="170"
defaultValue="Images/noPhoto.gif" />
```

Note that the only two valid attributes for storing photo data are the *jpegPhoto* and *thumbnailPhoto* attributes. Note also that we will re-render the photo the user provides so that the it is 130 pixels wide by 170 pixels high; this is configurable in the photo tag.

To enable URL support for the photo field, you need to change two options in the photo tag; you need to set the **type** and **attribute** options. Set *type="URL"* and set *attribute="wwwHomePage"*; in this example we are going to store the URL to the web page (wwwHomePage) attribute.

```
<photo label="Add/Change Photo" type="URL" visible="yes" editable="yes"
attribute="wwwHomePage" width="130" height="170"
defaultValue="Images/noPhoto.gif" />
```

Do we have a recommended attribute to store the URL? This is really up to you and how you plan to use the data. If you have an application (such as SharePoint) that is going to extract this URL and use it, then make sure it can access the attribute you are using. The ***wwwHomePage*** and ***url*** attributes are probably the two best choices; both of these are also accessible via Active Directory Users and Computers.

> **Note:** Directory Update performs no validation to ensure the URL path is valid. This allows you to use this feature exactly as you may require, but the user must use the path format you specify, such as http://servername.corp.local/pictures/user122.jpg

You need to remember to customize the Photo Upload dialog box labels. The default dialog box labels are shown in Figure 7, but these labels assume that you are using the type="file" option.
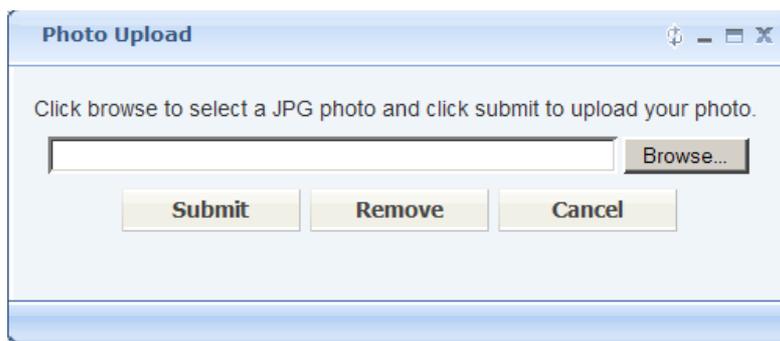
Figure 7: Photo Upload dialog box

The labels for the photo upload dialog box are configured in the AppSettings.XML file; the relevant section is shown Figure 8.

```
63        <!-- Photo upload window -->
64        <photoUploadWindow title="Photo Upload">
65          <instructions>Click browse to select a JPG photo and click submit to upload your photo.</instructions>
66          <validations>
67            <requiredFileValidation errorMessage="You have not select a photo to upload." />
68            <fileFormatValidation    errorMessage="The file was not a valid jpg file." />
69          </validations>
70          <buttons>
71            <submitButton text="Submit" />
72            <cancelButton text="Cancel" />
73            <removeButton text="Remove" />
74          </buttons>
75        </photoUploadWindow>
```
Figure 8: Photo Upload dialog box text labels in the AppSettings.XML file

Auditing can include date/time, user name, and/or IP address

For the past few versions, Directory Update has supported the ability to log the last date and time that Directory Update was used to update the user. This information is written to the attribute of your choice. By default, we use *extensionAttribute11*, but you can change this to any valid attribute; you may need to do this if your Active Directory forest has not been prepared to support some version of Exchange.

To enable auditing, edit the AppSettings.XML file and location the <auditing…> tag. An example of this is shown in Figure 9.

```
55   <!-- Auditing - Directory Manager will log the date in yyyy-mm-ddThh:mm:ss format format to the specified extension attribute -->
56   <!-- each time a user updates their information.  You can specify extensionAttribute1 through extensionAttribute15 -->
57   <!-- In order to use the extensionAttribute attributes, your Active Directory must have been prepped for Exchange 2000/2003/2007. -->
58   <auditing enabled="no" attribute="extensionAttribute11" showUserLastUpdate="yes" text="Your last update was">
59     <lastUpdateDateTime  audit="yes" />
60     <lastUpdateUserName  audit="yes" />
61     <lastUpdateIpAddress audit="yes" />
62   </auditing>
```
Figure 9: Auditing configuration in the AppSettings.XML file

The audit feature can also be useful if you need to run a script during logon to check and see if the user's information is up-to-date. We have an unsupported Auto Launch script on our web site that will check this attribute and automatically launch Directory Update if the user's information has not been updated in a few months old.

Adding your own attribute names

Though we would like to think we include most all attributes that any organization would want, we are frequently asked if the Directory Update administrators can add their own attributes. In v1.7 we now provide limited support for additional attributes. The primary restriction is that the attribute must be a Unicode String attribute; attribute types such as Boolean, Octet String, Integers, etc…

To add a new attribute to the interface, there are two files that need to be modified. These are the Default.aspx file that is found in the root folder of Directory Update and the DirectorySettings.XML file that is found in the Settings folder. You need to define the new attribute in both of these files and you must take care to put the new identifying information in to the correct section in both files.

In the following example, I will add the *carLicense* attribute to the interface and put it in the Organization section and just below the Department Number attribute. First, I have to edit the DirectorySettings.XML file and add a new line below the <departmentNumber…> tag. In the example shown in Figure 10, you can see a new line for the carLicense attribute. The tag identifier must be the LDAP attribute name of the attribute.

```
  </department>
  <departmentNumber label="Department #" type="text" visible="no" editable="yes"></departmentNumber>
  <carLicense label="License Plate #" type="text" visible="yes" editable="yes"></carLicense>
  <title label="Title" type="dropdown" visible="no" editable="yes">
```
Figure 10: Adding a new line to the DirectorySettings.XML file

In the example in Figure 5, I simply copied the line above and then pasted in it. Remember to change both the open tag and close tag components, e.g. *<carLicense….></carLicense>*

The next step is to edit the Default.ASPX file; this file is a bit more complicated than the DirectorySettings.XML file, but it has the same sections. In this example, I need to locate the Department Number section. Each attribute has 2 lines, a comment line and the actual <uc:CommonField…> tag. An example is shown in Figure 11.

```
<!-- Department Number -->
<uc:CommonField ID="departmentNumber" runat="server" Visible="false" MaxLength="64" />
<!-- License Plate -->
<uc:CommonField ID="carLicense" runat="server" Visible="false" MaxLength="64" />
<!-- Title -->
<uc:CommonField ID="title" runat="server" Visible="false" MaxLength="64" />
```
Figure 11: Adding the field to the Default.ASPX file

In this example, I simply copied these two lines and then pasted them below the existing lines. Then I modified the ID parameter to reflect the attribute *carLicense*.
```
<!-- Department Number -->
<uc:CommonField ID="departmentNumber" runat="server" Visible="false" MaxLength="64" />
```

The result is that a new attribute for the *carLicense* attribute (shown in Figure 12 as the **License Plate #** label).

Company	Company 6	Office	Office 2
Department	Engineering	License Plate #	Hawaii - JMW-747
Manager	Jim McBee		

To search for a manager, type the first few characters of the person's first name and select the correct name from the list.

Figure 12: New attribute now on the user interface

[Note: The v1.5 release notes are included for reference]
Release Notes v1.5

We have re-branded our company and now operate completely independently as Ithicos Solutions (http://www.ithicos.com). We are the same people as before, but with a new name.

We are pleased to announce the release of Directory Update v1.5. Directory Update v1.5 introduces many small improvements that our customers have been requesting and builds on our existing features from earlier versions. The following are features that have been introduced in v1.4 and v1.5:

- Address Sets feature allow a user to select a specific attribute (such as office) and have address (street, city, state, postal code, country, post office box) automatically filled in.
- New XML configuration files should simplify configuration.
- XML settings files moved to subfolder called Settings.
- Multi-domain support - One instance of IIS can support multiple domains in the same Active Directory forest.
- Improved localization by changing all labels shown on interface, even buttons.
- Enable / disable displaying only Exchange-enabled user accounts in the Manager, Assistant, or Secretary lookup fields.
- Ability to provide additional filtering of names that will show up in the Manager, Assistant, or Secretary lookup fields.
- Telephone number format validation using regular expressions (RegEx)
- License key for v1.5 is different than earlier versions.
- Enable or disable global catalog server support for Manager, Assistant, and Secretary lookup fields - Multiple domain users should enable this feature.
- Display note or disclaimer on logon screen.
- You can now make any field required; meaning it must have a value.
- You can now set a default value for a field; if the field is empty in Active Directory then default value is automatically filled in.
- Installer no longer requires service account to be a member of Domain Admins or Account Operators to proceed.
- Each section of the Directory Update interface now has a "notes" section.
- Settings Updater (for updating license key, service account, domain, and domain controller information) has been renamed to Configuration. It can be found on the All Programs -> Directory Update menu.
- Parent Registry key changed to Ithicos to reflect new company name.
- Product namespace changed to Ithicos - this only shows up if the program crashes on something, though.
- Miscellaneous bug fixes

More on these new features can be found later in the release notes. We encourage you to read more.

Installing Directory Update v1.5

Installing Directory Update v1.5 is very similar to installing previous versions. However, the AppSettings.xml file from previous versions is not compatible with v1.5. Once you install the new version, you will need to make your custom configuration changes to the new XML files and Style.css file if applicable.

You can take your drop-down list values from your previous versions and copy them to the new DirectorySettings.xml file, though. Here are the steps we recommend to upgrading:

1. Make a backup copy of your c:\inetpub\wwwroot\directoryupdate folder (or where ever you installed the software originally.
2. Run Control Panel -> Add/Remove Programs and remove the previous version of Directory Update
3. Delete the Registry key HKLM\Software\ITCS Hawaii
4. Delete the directory c:\inetpub\wwwroot\directoryupdate (or where ever you installed Directory Update originally.)
5. Install the new version of Directory Update (using the DirectoryUpdate.msi installer)
6. Copy your customized <value> </value> tags from the original AppSettings.xml file and add them to the new .\Settings\DirectorySettings.xml file
7. Make additional customizations to the AppSettings.xml and DirectorySettings.xml files such as changing labels, hiding fields or making fields required.

The trickiest part of the upgrade is moving your existing <value> tags to the new DirectorySettings.xml file. Here is an example of the old AppSettings.xml file in which the Title field has been changed to a "dropdown" list, made visible and editable.
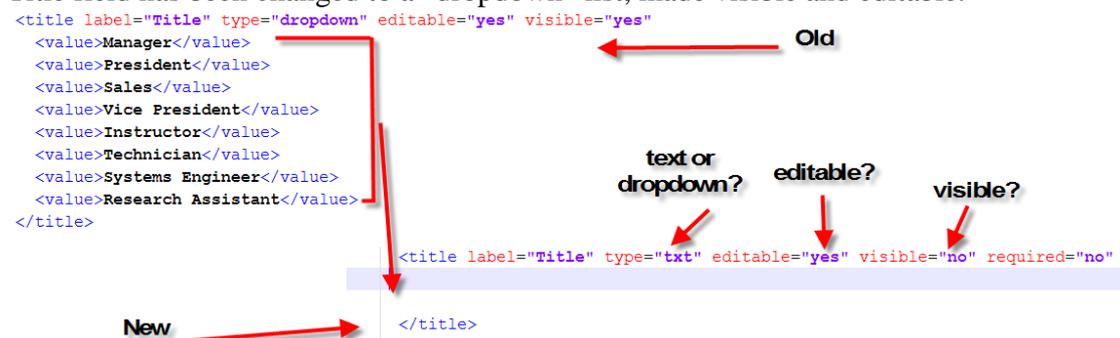
In order to convert this data, there are a couple of things you would need to do in the DirectorySettings.xml file.

1. Make sure that the option type="dropdown" is set
2. Make sure that the option editable="yes" is set
3. Make sure that the option visible="yes" is set

4. Copy all of the <value> … </value> tags from the earlier version's AppSettings.xml file inside of the <title> … </title> tags.

This is much easier using an XML editor that will highlight tags, comments, and help you with the open/close tags.

Notepad ++

If you are still editing your XML files using a boring editor like Notepad, we strongly urge you to download the free Notepad ++ editor. It will make editing our XML files much easier because it puts everything in color and it makes common errors such as not closing a tag much easier to find. You can find Notepad++ at http://notepad-plus.sourceforge.net/uk/site.htm

New XML Configuration Files

In an effort to make configuration a bit simpler and also more compatible with our other products, we have separated the configuration in to 3 separate configuration files. These are all found in the \inetpub\wwwroot\DirectoryUpdate\Settings folder.

- **AppSettings.xml** is the master configuration file for the application. From the AppSettings.xml file, you can
    o Localize the buttons
    o Localize the form/window labels
    o Customize your help messages and links
    o Specify search filers for the Manager, Assistant, and Secretary fields
- **DirectorySettings.xml** file is the master configuration file for the fields that the user sees on the User Information form. From this file, you can:
    o Hide/show fields
    o Change fields to drop-down lists or text boxes
    o Set a field to be required
    o Set a default value
    o Specify values for the drop-down lists.
- **AddressSettings.xml** is the file that holds Address Sets data. From this file, you can enable Address Sets and enter information that corresponds to one of your drop-down lists in the DirectorySettings.xml file.

Address Sets

Address Sets is one of the most exciting and most requested features in Directory Update 1.5. A user can select a drop-down list value in a field such as office and have their address information automatically filled in. The following fields can be used as the control field for an Address Set: Company, Office, Division, Department, and Department Number. The default control field value is the Office field.

To enable Address Sets, edit the AddressSets.xml file and set the enabled option to enabled="yes" in the <addressSettings…> tag.

```xml
<addressSettings version="1.5" controlField="office" enabled="yes">
  <addressSetting value="Office 1">
    <streetAddress>Street Address 1</streetAddress>
```

Next, in the DirectorySettings.xml file file, choose the field you want to use, configure the type to be type="dropdown" and then put in values for each address set you want to use. For example, if you are using the Office field, you must have offices for each address set you want to use.

```xml
<office label="Office" type="dropdown" editable="yes" visible="yes" required="no" defaultValue="">
  <value>Office 1</value>
  <value>Office 2</value>
  <value>Office 3</value>
  <value>Office 4</value>
</office>
```

Next, in the DirectorySettings.xml file, you will probably want to set each of the address fields so that it is visible, but it is set so that it is NOT editable. This will ensure that the selection that the user makes from the address set is visible to the user, but so they cannot try and edit it.

```xml
<!-- ADDRESS SECTION -->
<address label="Address" visible="yes">
  <streetAddress label="Street Address" type="text" editable="no" visible="yes" required="no" defaultValue="'
  <roomNumber    label="Room #"        type="text" editable="no" visible="no"  required="no" defaultValue="'
  <postOfficeBox label="P.O. Box"      type="text" editable="no" visible="no"  required="no" defaultValue="'
  <city          label="City"          type="dropdown" editable="no" visible="yes" required="no" defaultValu
    <value>City 1</value>
    <value>City 2</value>
    <value>City 3</value>
    <value>City 4</value>
  </city>
  <stateOrProvince label="State/Province" type="dropdown" editable="no" visible="yes" required="no" defaultVa
```

Finally, for each value in the drop-down list (such as office), you must enter valid address information for that selection in the AddressSets.xml file. Below is an example for Office 1 (where the Office field is the control field).

```xml
<addressSettings version="1.5" controlField="office" enabled="yes">
  <addressSetting value="Office 1">
    <streetAddress>Street Address 1</streetAddress>
    <roomNumber></roomNumber>
    <postOfficeBox></postOfficeBox>
    <city>City 1</city>
    <stateOrProvince>Alaska</stateOrProvince>
    <zipOrPostalCode>11111</zipOrPostalCode>
    <country>US</country>
  </addressSetting>
  <addressSetting value="Office 2">
    <streetAddress>Street Address 2</streetAddress>
```

Note that even if you do not use the Room Number or Post Office Box fields, the tags **must** be inside of each address setting tag.

Telephone Number Format Validation

A fairly common request from customers is to make sure that when a user enters a phone number that the phone number is in the correct format. For example, in the U.S. you might want the area code to be in parenthesis and the exchange to be separated from the rest of the number by a dash, such as this format: (808) 555-1234.

We now do allow you to enable phone number format validation. You must specify the phone number format using a regular expression. You enable the phone number format on a per-phone number basis (in case each phone number needs to be a different format.) First, though, you must define the phone number format. This is done in the DirectorySettings.xml file. Each format allows you to specify an example that the user will see if they type in the wrong phone number format. The default DirectorySettings.xml file has a few examples.

```
<!-- Telephone number format validation uses regular expressions. Each phone number field can use a different validation format. -->
<validations>
    <validation format="USA"       formatExample="Example: (808) 123-4567 or 808-123-4567 x4321" regularExpression="((\(\(\d{
    <validation format="UK"        formatExample="Example: +44 (0) 123 456 7890 or +44 (0) 1234 567890" regularExpression="
    <validation format="Australia" formatExample="Example: +61 2 1234 5678 or +61 (0)3 1234 5678" regularExpression="(\+61
    <validation format="Norway"    formatExample="Example: +47 12 34 56 78" regularExpression="^(\+47 )(\d{2} )(\d{2} )(\d{
</validations>
```

Once you have defined the format you want, for each phone number, locate the validationFormat option and specify the format name. For example, if you want to use the existing USA format, you would specify validationFormat="USA".

```
visible="yes" required="no" validationFormat="USA" showExample="no" defaultValue=""></officePhone>
visible="no"  required="no" validationFormat="" showExample="no" defaultValue=""></otherOfficePhone>
visible="yes" required="no" validationFormat="" showExample="no" defaultValue=""></fax>
```

For non-programmers, REGEX's are not simple. However, you can define a formula for checking just about anything, if you have the patience. Here is some places to check for more information:
http://en.wikipedia.org/wiki/Regular_expression
http://www.regular-expressions.info

Multi-Domain Support
In earlier versions of Directory Update, multi-domain support was a bit of a pain in the neck. Directory Update needed to be installed on a separate IIS server for each domain. We have addressed that issue and now you can configure Directory Update to support multiple domains from a single Windows 2003 IIS server. There are a few prerequisites:
- All domains you will connect to must be in the same Active Directory forest
- You must have a user account in each domain that you can use as a service account or you can use a single service account but it must be assigned rights to update users in all domains.
- You must know the name of a domain controller / global catalog server in each domain
- We recommend that the IIS server be on a low latency connection (less than 200ms) between each of the domain controllers it will be connecting to.

To use the multi-domain support, install Directory Update just as you normally would and configure the first domain. Then run the Configuration utility found in All Programs

-> Directory Update -> Configuration. From there, choose Add A New Domain Instance and click Next. Configuration the additional domain is just like configuring the first domain. You will need the following information:
- Domain controller / global catalog server name
- DNS domain name
- Service account name
- Service account password
- Organization / company name used to register the software
- License key generated for the additional domain (or use the evaluation mode)

Lookup Field Filtering

The Directory Update interface has three lookup fields, these are the Manager, Assistant, and Secretary fields. Usually the most common of these fields is the Manager field.

Employee ID	BR-549		Employee Type	Full Time	▼
Manager	Bob Reinsch	🔍	Assistant	Ben Craig	🔍
To search for a manager, type the first few charactors of the person's first name and select the correct name from the list.					

Active Directory stores the data in these fields as a distinguished name rather than simple text. This means that the Manager, Assistant, or Secretary **must** exist in the Active Directory. In earlier versions of Directory Update, our query for valid users provided a listing of all mail-enabled users, but not users without mailboxes. Further, we query a global catalog server for this listing rather than the domain controller port.

We are now providing the ability to use a domain controller rather than the global catalog server. In the <lookupFields…> tag set the useGlobalCatalog to ”no”. If your organization does not use Exchange server or if not all of your users are mailbox-enabled, then set the showOnlyExchangeEnabledUsers option to "no".

```xml
<!-- Use the Global Catalog only if you have more than one domain that has user accounts. -->
<!-- ShowOnlyExchangeEnableUsers (if enabled) will display only users with mailboxes. -->
<lookupFields useGlobalCatalog="yes" showOnlyExchangeEnabledUsers="yes" maxResults="20" />
<!-- Allows you to specify accounts that will NOT be displayed Manager, Assistant, or Secretary lookup fields. -->
<accountFilter enabled="yes" attribute="extensionAttribute12" value="excluded" />
<!-- Auditing - Directory update will log the date in YYYYMMDD format to the specified extension attribute -->
<!-- each time a user updates their information. The Exchange 2000/2003/2007 forest setup process must -->
<!-- have been run to use this feature since the extension attributes are only created then. -->
<!-- You can specify extensionAttribute1 through extensionAttribute15 -->
<auditing enabled="yes" attribute="extensionAttribute11" />
```

One request that we have seen a number of times with previous versions is that when a user tries to lookup someone in the Manager, Assistant, or Secretary they will often see system user accounts, administrators, or service accounts. We are letting you specify a filtering mechanism, by default extensionAttribute12 is used, but you can use any attribute. We chose a value of *excluded* but you could also change that to something else.

Note that if you enable the showOnlyExchangeEnabledUsers=”yes” option, any user account whose mailbox is hidden from the Global Address List will be hidden from the results list as well.

Improved Localization or Interface Text Customization

We are pleasantly surprised at the number of international customers that we have. Naturally, companies that use languages other than English would like to customize the interface as much as possible. We have two types of localization or re-labeling that you can do. The first is on the portion of the interface that the user actually works with. These include things such as the actual field labels, phone number examples, and the notes that show up at the bottom of each section. These are customized in the DirectorySettings.xml file.

In the graphic below, you can see where the field labels, section notes, and the phone number examples.

The AppSettings file contains the remainder of the interface customization settings. Below shows an example of customizing the page title, logon credential prompts and logon error prompts.

The following are the options that can be customized or localized using the AppSettings.xml file:

- Web browser page titles
- Logon prompts for user name, password
- Logon error message (when use enters invalid user name and password)
- Domain name drop-down title and whether or not it is visible
- Forms-based authentication logon page

- Help button text and link
- Button labels
- Update confirmation text
- Warning and error message text

Configuration Utility

In previous versions, we used a utility called the Settings Updater that allowed you to edit your license key, change the domain controller/global catalog or to change the service account. This utility has now been renamed to just Configuration and can be found at All Programs -> Directory Update.

From this utility, you can
- Add additional domains in a multi-domain forest
- Modify the service account for any instance of Directory Update
- Change the service account password for any instance of Directory Update
- Change the domain controller or global catalog server

New Registry Key Location

We have moved all of our Registry information in to HKLM\Software\Ithicos. Customers rarely need to know this, but we like to disclose where we store information in the Registry just in case.
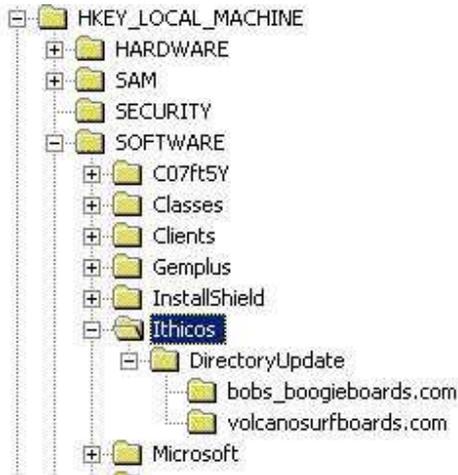
Logon Disclaimer / Note

A few customers have asked for the ability to put in a disclaimer or a note on the forms-based authentication logon page. This is configured in the AppSettings.xml file.

```xml
    <note>
        <!-- Logon page disclaimer or help information. -->
        <message>© 2007 - Ithicos Solutions. All rights reserved.</message>
    </note>
</logInPage>
```

Bugs fixed

We have found a few issues that sometimes affected users or that sometimes generated errors in the event logs. Here are some fixes to the code:

- We found a few places in the code were processes we not being closed out properly if a user just closed the browser without doing anything the IIS Worker process would terminate and restart. This exception is now being handled.
- Sometimes connecting to a global catalog server was problematic so we give users the option of turning this on or off. It is only necessary in multi-domain networks.
- Organizations that do not use Exchange as their mail system are now being accommodated better.
- Additional filters have been included so that your users do not see service accounts or special mailboxes in the lookup fields.
- If you use the Assistant field, when you choose an assistant, we automatically populate the Exchange GAL assistant field with the assistant's display name. The Active Directory Assistant attribute is not directly tied to that attribute.