XEROX®

FreeFlow™

Version 6.0, September 2007
701P47190

FreeFlow®
Web Services
LDAP Integration

Powered by Press-sense

Xerox Corporation
Global Knowledge & Language Services
800 Phillips Road
Building 845
Webster, New York 14580

While every care has been taken in the preparation of this material, no liability will be accepted by Xerox Corporation arising out of any inaccuracies or omissions.

Printed in the United States of America.

Companies, names, and data used in examples herein are fictitious unless otherwise noted.

Changes are periodically made to this document. Changes, technical inaccuracies, and typographical errors will be corrected in subsequent editions.

Contents

Overview 1

Prerequisites 1

Workflow Description 2

Method Configuration... 3

Mandatory Parameters 3

User Attributes 5

Attribute Element Detailed Description 5

Reference Element Detailed Description...5

Privilege Mapping... 6

Backward Compatibility.. 6

Overview

The *FreeFlow Web Services Lightweight Directory Access Protocol (LDAP) Adapter* enables user authentication when logging in to the *FreeFlow Web Services* application by way of a directory service.

Note: *Microsoft Active Directory* and *Novell's NDS* are currently supported by *LDAP*.

Integration of the *LDAP* system into *Web Services* will enable users who are managed in an *LDAP* directory to work with the *Web Services* system. These users can log into *Web Services* and be seamlessly authenticated against the *LDAP* directory. In addition, user information can be automatically updated from the *LDAP* directory into *Web Services*.

Integration of the *LDAP* system into *Web Services* is intended for corporate Print Buyer accounts and supports multiple *LDAP* directories for different customers.

Prerequisites

Use of the *FreeFlow Web Services LDAP Adapter* requires the following two preconditions:

- Appropriate *Directory User Account(s)* should be set up and made available
- The *Web Services* application should be properly configured to perform *LDAP* authentication

In addition:

- The user must be able to log in to the *Web Services* by providing domain user account credentials
- The *Web Services* user account must be synchronized with the domain user; including account state, required user details and properties.
- *Web Services* must be able to recognize and process accordingly to the particular error conditions

Note: There are a number of predefined error states:

- Authentication failed – wrong login name or password has been provided by user during login.
- Account disabled or deleted – previously active directory user account has been disabled or deleted.
- Technical error – any other technical error occurred during authentication (network problem, *LDAP* server error, directory access denied, etc).
- Application error – any other application error occurred during authentication (missing configuration parameters, format error, etc).

Workflow Description

The following three steps describe the *LDAP* workflow:

1. The user enters the login URL that will provide external (*LDAP*) authentication: http://localhost/iway/?IID=xxx (where xxx is the authentication identifier as defined in the configuration file. For more details, see the *Method Configuration* paragraph).
2. The user enters the domain login name and password in order to log in to the *Web Services* application; according to configuration parameters, *Web Services* performs user authentication on *LDAP* server.
3. If authentication is successful, the user will log in to the system. In this case:

   i. If no corresponding *Web Services* user account exists, the system will create an account using retrieved domain user details and properties.

   ii. If the corresponding *Web Services* user account already exists, the system will update the account using retrieved domain user details and properties. If the corresponding *Web Services* user account was disabled, the system will enable the account.

   iii. If authentication fails, the user will not log in to the system and the appropriate error message will appear. In this case:

      - If the domain user account has been disabled or deleted but the corresponding *Web Services* user account still exists and is active, the system will disable such account.
      - For all other error states (wrong password, technical error, etc.) no other manipulations on the corresponding, existing *Web Services* user account will be performed.

Method Configuration

Necessary configuration parameters should be setup in the /Newedition/IPanel/Integrations/ExternalMethods/_nw_setup.xml configuration file.

Each configuration has its own section by defining configuration ID. For example, it may be possible to define more than one *LDAP* server on which different users may be authenticated. In this case, two different configurations should be present in the configuration file – one for each *LDAP* server:

```
<root>
        <externalIntegration IID=”1”>
                …
        </externalIntegration>

        <externalIntegration IID=”2”>
                ….
        </externalIntegration>
</root>
```

In order to use the appropriate configuration, the user should provide a configuration ID in the login URL (see above): http://localhost/iway/?IID=1

Mandatory Parameters

<method> - defines authentication method that should be performed. For *LDAP*, the value always should be “LDAP” :

```
        <method>LDAP</method>
```

<serverType> - defines directory service (*Microsoft Active Directory* - “*AD*” or *Novell NDS* - “*E*”).

```
        <serverType>AD</serverType>
```

<authURL> - defines *LDAP* server name or *IP* address.

```
        <authURL>10.10.10.1</authURL>
```

<searchBase> - defines the base path to search in directory and consists of a domain name and zone. For example, mydomain.com should be configured as:

```
        <searchBase>DC=mydomain,DC=com</ searchBase >
```

<ldapPort> - defines the *LDAP* server port number (default 389)

```
        <ldapPort>389</ldapPort>
```

<filter> - defines filter for directory search procedure. This should not be changed. Always use predefined value:

```
        <filter>(&amp;(objectClass=User)(sAMAccountName={0}))</filter>
```

<adminUsername> and **<adminPassword>** - define domain administrator credentials. In order to access the directory, and perform search and data retrieve procedures, these parameters must be provided. In general, this should be the login name and password of any user who belongs to the "Domain Admins" group.

<domain> - defines the domain name

<domain>mydomain</domain>

<customerID> - defines a customer ID under which the corresponding user account should be created (if a corresponding user account does not already exist). For example: first time login will cause a user account creation under a specified customer. Must be an existing customer ID.

<customerID>2</customerID>

<attributes> - defines a set of user attributes and properties that should be retrieved from the directory entry to update the *Web Services* user account. There is a predefined set of attributes that can be used. To avoid using particular attributes, unnecessary attributes should be commented (the entire corresponding "attribute" element, including sub-elements, if any). For more details, see the available *User Attributes* table below.

Note: There are two methods by which directory entry attributes can be used in the *Web Services* user account:

- Exact mapping – the value of the directory entry attribute will be used as is in the *Web Services* user account.
- By Value mapping – *Web Services* corresponding attributes accepts a specific value that depends on what is defined in the directory.

User Attributes

Directory Attribute	Web Services attribute	Description	Mapping method
givenName	FirstName	User’s first name	Exact
sn	LastName	User’s last name	Exact
streetAddress	Street	Street address	Exact
l	City	City	Exact
st	State	State or Province	Exact
postalCode	ZipCode	Zip/Postal code	Exact
co	Country	Country	Exact
telephoneNumber	Phone	Phone number	Exact
mobile	Mobile	Mobile number	Exact
facsimileTelephoneNumber	Fax	Fax number	Exact
mail	Email	E-Mail address	Exact
company	CompanyName	Company Name	Exact
title	JobTitle	Job Title	Exact
memberOf	PrivilegeID	Privilege level	By Value*

* - see privilege mapping section.

Attribute Element Detailed Description

Field name	Description
Name	The name of directory user attribute
refTo	The name of *Web Services* user attribute
Type	Attribute type in the directory
refMethod	How directory entry attributes can be used in the *Web Services* user account

Reference Element Detailed Description

Field name	Description
attrValue	The value of directory user attribute
refValue	The value which *Web Services* user attribute should accept if directory user attribute has value defined in attrValue field.

Privilege Mapping

If the *Directory User* belongs to any group(s) with administrator rights (Domain Admins, Enterprise Admins, etc), the corresponding *Web Services* user account should have Admin privileges. Setting Admin privileges can be carried out by mapping a "memberOf" *LDAP* user attribute with a value that matches "Admin" or "Admins" in the *Web Services* "Admin" privilege level. In other words, if the *LDAP* user is a member of group(s) whose name(s) contain(s) "Admin" or "Admins", such a user will receive the Admin privilege level on the *Web Services* side. All other user groups will accept "SuperUser" privileges on the *Web Services* side.

Important: It is essential to maintain a "Top – Down" order in the directory groups to the *Web Services* privilege level mapping. A group that has the highest rights in the directory should be defined first. The last reference element should contain a group with minimal rights.

```
<attribute name="memberOf" type="text" refTo="PrivilegeID" refMethod="byValue">
                        <reference attrValue="Admins" refValue="Admin"/>
                        <reference attrValue="Admin" refValue="Admin"/>
                        <reference attrValue="Users" refValue="SuperUser"/>
</attribute>
```

Backward Compatibility

This *LDAP* implementation version will not be backward compatible with the pre-3.1 *LDAP* implementation version:

- Re-configuration will be required on the system
- Previous users who were associated with *LDAP* will no longer be associated with *LDAP*. The next time that previous users log in to their profiles, they will be required to be re-created from scratch, without any backward compatibility