



UnitySync® Administrator Guide

Directory Wizards Inc

February 5, 2025

Contents

About The UnitySync Administrator's Guide	3
UnitySync Overview	4
Planning Your UnitySync Installation	8
Prepare Directory Servers for Use With UnitySync	10
The UnitySync User Interface (UI)	17
General Tab	19
Source Tab	24
Destination Tab	31
Display Name Tab	40
SMTP Addresses Tab	43
Custom Tab	47
Email Notify Tab	49
Back-up and Restore of UnitySync	51
Move or Reinstall UnitySync	52
Uninstall UnitySync	54
Troubleshooting	55
Appendix	57
Appendix 1: Trim DN	58
Appendix 2: Custom Mapping	60
Appendix 3: Filters & Exclusions	76
Filters Office 365 (O365)	77
Appendix 4: Automating your UnitySync Connections	79

About The UnitySync Administrator's Guide

What's in this guide?

The UnitySync Administrator's Guide will help you:

- understand UnitySync and its features
- configure UnitySync and customize features specifically for your environment
- synchronize Users, Groups and other person objects
- troubleshoot issues with UnitySync

What's not in this guide?

Since we'll be going into detail, we've found that it's often easier to follow along with a basic UnitySync installation. The basics of getting started are in the UnitySync Release Notes & Install Guide, found at www.dirwiz.com/UnitySync/doc.

Our advice? Take a look at the Install Guide, get UnitySync installed and started, then come back to this guide for a more in-depth look at UnitySync.

Who is this guide for?

While it is intended for system/directory administrators who are already familiar with LDAP directory servers, it's also for anyone who wants to learn about and give UnitySync a try. If any concept seems confusing, please contact us at support@dirwiz.com for clarification.

How to use this guide

This guide outlines the numerous features UnitySync has to offer. If you're a seasoned user, refer to it when you have a problem to solve or wonder if you can tweak UnitySync to make your job easier. For those new to UnitySync, read or skim through for an overview of what UnitySync can do and feel free to contact us for a demo, help with your initial configuration, or to assist you in tackling complex requirements.

IMPORTANT NOTE: *Keep an eye out for notes in this format. When you see them, we're alerting you to an important bit of information.*

How to contact us

The best way to reach the most of us at once is by emailing support@dirwiz.com. We can assist you via email or schedule a call or screen share if needed.

Refer to this knowledge base article for information on what we need to best assist you when you're having a problem. Visit our Support page for other support options.

Have an idea for a new feature? Is there something you wish UnitySync could do for you? If you've had a lightbulb moment for new features, products or even tiny utilities, let us know! Our development team is very responsive to your needs.

UnitySync Overview

About UnitySync

UnitySync is the culmination of many years working with email directory environments. UnitySync is a Web-based directory synchronization application that connects 2 or more directories, and syncs data between them. After a synchronization run, end users see email addresses, contact information, etc. from other directories within their own directory address books.

UnitySync is not itself a directory. It only stores configuration data, log files, an intermediate LDIF file, and a small SQL database file to track deltas. In essence, UnitySync stores only what it needs to run a synchronization.

How UnitySync Works

UnitySync runs on a web server, allowing users to administrate their syncs using a web browser. Each configured connection will read from one (Source) directory and write to another (Destination) directory. Create the same connection in reverse to get a complete 2-way synchronization run. In its very first execution, each UnitySync connection does a full directory read, synchronizing whatever combination of objects you've asked it to synchronize. On subsequent synchronization runs, UnitySync reads the entire Source directory but only writes changes (deltas) to the Destination directory.

Configuration of each connection requires the input of directory information including IP, User ID and Password for both the Source and Destination directories. You will also specify which Source objects to pull (the default is all) and where to create the new contacts on the Destination.

The default configuration will create mail enabled contacts in your Destination directory using options and attribute mappings included in the default map template. You may further customize your UnitySync connection to take advantage of features that allow you to manipulate or organize your synced data to meet your unique requirements. These optional features are easily configurable within each connection. commonly implemented customizations include Display Name formatting, structure creation, Join with Existing Objects, Swap Proxy, and Custom attribute mapping.

You may download a fully functional copy of UnitySync from our Download page. You will receive a temporary key to get started. You may contact Technical Support Team to extend your evaluation or with any questions you have as you evaluate and test UnitySync in your own environment.

Features and Benefits

- can be run with one central copy or with multiple copies run by local directory managers
- can be centrally configured and then easily distributed for local execution
- can unsync a synchronization run if needed
- works with Microsoft Exchange/Active Directory, Amazon Simple AD, Azure, Office365 (O365), ADAM/ AD LDS, Google Workspace, OpenLDAP, CSV files, LDIF files, ODBC databases, ViewDS and more.

UnitySync Server System Requirements

Below are the complete UnitySync System Requirements (as of v4.6).

IMPORTANT NOTE: *There are additional special requirements for Office 365 (O365), Azure, and use with CyberArk linked at the end of this article.*

Operating System

- Windows
 - 2019 (64 bit) or better
- Linux
 - 64 bit Operating System
 - glibc 2.28 or greater - examples:
 - * Debian 10
 - * Red Hat Enterprise Linux 8

- * Fedora 29
- * Ubuntu 20 LTS
- * OpenSUSE Leap 15.2

IMPORTANT NOTE: *If you are unsure of your glibc version, please see Finding Your glibc Version.*

Web Server/Browser

- UnitySync must either be installed on a web server (i.e. IIS, Apache) or used with the included dirweb web server utility
- You will access the UnitySync user interface via web browser - please use Microsoft Edge, Firefox 6 or Chrome 6 or newer. (Microsoft Explorer is no longer supported).
- Browser must have javascript enabled

General System Requirements:

- Hard Disk Space
Minimum: 500Mb
Recommended: >10Gb

IMPORTANT NOTE: *Drive space utilization varies depending on the number of objects synced and what level logging is regularly used. Default Log File setting is 2-Lowest and this is the level we recommend for regular use. Consistently logging at higher Log File levels will require more disk space. **Low disk space can cause serious issues.** Use log file management to purge log files regularly.

Memory Usage

- **Minimum:** Minimum recommended OS memory recommendation
- **Recommended:** Minimum practical OS memory recommendation

CPU

- **Minimum:** 2Ghz Processor
- **Minimum:** Dual or Quad processors

The larger the sync environment, the more spokes off your hub, the more objects syncing - all of these things can affect performance. We recommend increasing the number of processors as these parameters increase. Alternatively, multiple installs of UnitySync can be used to spread the load.

Network

- TCP/IP Connection
- Access thorough firewalls to ldap ports (389, 3268, ssl 636 3269)
- Optionally, VPN may be used
- The UnitySync server must be configured with a DNS server and a DNS Host entry created.
- SMTP server (optional)

User Account/Password

- A User account (with both read/write permissions) for each DC being synced.
- Ideally, a Domain Admin will be used. However, non admin and/or read only permissions may be used in a decentralized environment.

Special Requirements

- Office 365 (O365) requirements
- Azure requirements

- CyberArk requirements

IMPORTANT NOTE: *Your results may vary! They are often dependent on unique factors in your environment; the size of the data, communications speeds and other variables outside of the scope of testing. Contact our Technical Support Team if you have any questions regarding your installation and use of UnitySync.*

How is UnitySync licensed?

Licensing

UnitySync licensing is based on:

- Your initial installation of our software; we call this your base.
- How many additional installations of UnitySync your solution requires; this is often a lab or test copy.
- How many target directories you are syncing; a minimum of 2.
 - Each LDAP directory, counted by unique configured IP, counts as one directory license.
 - Likewise, each ODBC DSN or o365 tenant counts as one directory license.

Each directory need only be licensed once. Each may be used as a Source and/or Destination of any connection. There is no limit on the number of connections configured between each licensed directory. There is no limit on the number of objects synced between directories.

Your key and installation

The unique Install Serial Number as it appears on the User Interface (UI) Launch Page is used to generate a license key tied to your individual installation, and includes an accounting of your purchased directory licenses. On the Launch Page, you will see:

- Your installed UnitySync version number
- The Release Date of that version.
- Your Install Serial Number. This will change with each installation.
- The License Key entry box. To apply your key, enter it here and click on Save Key.
- Your Key Info (once your key is saved).
- The number in parentheses after Configured Directories shows a total of how many IPs are actually included in ALL of your current connections.
- Below that, a complete list of all IPs (or hosts) included as a source or destination directory. If an IP appears, it must still exist in one of the connections.

Syncs will fail to run (throw a license key error) if the *Configured Directories* count exceeds your *Directories* license limit. You can fix this by deleting unused connections, or by editing the Source and Destination tab IP's so your number of configured directories stays within your licensed limit.

Maintenance Licensing

A current Maintenance License entitles a client to receive Technical Support as well as product upgrades as they become available.

The first year of maintenance licensing is included in your initial purchase. Maintenance licensing can be renewed yearly for a fee of 15% of the current purchase price.

What is included under my Software Maintenance Coverage?

With up to date Software Maintenance coverage you will receive personal, written or verbal answers to all technical questions. Responses to most inquiries are usually provided in the same business day, often within minutes to an hour of receipt. Answers come directly from an experienced DirWiz technical support specialist, involving members of our software development team whenever necessary. Support inquiries may include requests for help with any aspect of the software's implementation: installation, configuration, testing, troubleshooting. You may also request help to meet unique requirements, specific to your own implementation.

When new versions become available, software upgrades are free to all customers with current Software Maintenance coverage. Each new release includes enhanced functionality and updated security features.

Customers with current maintenance coverage may purchase additional software and directory licenses without having to repurchase the original installation.

Through ongoing interaction with end users, our software is always improving and enhancing its functionality. If existing functionality does not meet your requirements, you may make a request for enhancements or feature upgrades. DirWiz Technical Support will work directly with you and our Development team to understand your requirement and determine if an enhancement can be made available. Whenever possible, requested feature enhancements are developed and released for immediate implementation. Otherwise, new features will be available in the next released version.

Technical Support is available Monday - Friday, 9:00am - 5:00pm EST (excluding US Federal Holidays). Please allow up to 24 hours for response. While we respond to most inquiries within the same business day (and often within minutes of receipt) we will sometimes research your question thoroughly before responding, delaying an immediate response.

To make a technical support request, please contact DirWiz Technical Support or 302.482.8004 (option 2). Please note that it is often more efficient to reach out via email initially, providing detailed information per this knowledge base article, as it often takes some time to research solutions.

UnitySync Security

UnitySync supports LDAP directories with SSL installed and enabled. To include SSL in a synchronization run:

- go to the Source and/or Destination tab
- select the SSL option
- enter the SSL port number

The SSL port number is different than a standard LDAP port number. That is, the standard LDAP port number for Active Directory (AD) and Exchange 200x (Ex200x) is 389, but the standard SSL LDAP port number is 636. Security of the UnitySync Manager may be controlled via IIS virtual directory share.

UnitySync Installation and Upgrade Instructions

The current release of UnitySync is available for download at <https://dirwiz.us/download>. Fill out the form and you will automatically receive the software download link as well as a 15 day temp key to get you started.

Complete upgrade and/or installation instructions are included in the Release Notes - Install Guide. Make sure to completely read and follow instructions for your user type.

If you already own UnitySync, once you've installed and are running UnitySync, send the Serial Number as it appears on the main page of the UnitySync interface to keys@dirwiz.com. We can then generate a permanent license key for you.

If you are not yet a UnitySync client, please contact our Sales Team for your personalized quote today.

UnitySync User Types

Depending on which type of user you are, you'll follow one of the following instructions:

- First Time UnitySync-v4.x for Windows User
- First Time UnitySync-v4.x for Linux User
- Existing UnitySync User: For users currently running a previous UnitySync version

Planning Your UnitySync Installation

There is a bit of planning to do if this is your first time installing and configuring UnitySync. You'll need to answer a few questions to determine how UnitySync can best meet your needs. You may want to consider:

- How many directories will you synchronize?
- What type of directory/server environment do you have?
- Who will have what level of access?

Luckily, UnitySync's modular design is meant to be flexible enough to suit the needs of your specific environment no matter what they are. Here are some general examples to illustrate pre-installation planning, directory environment situations, and their solutions.

IMPORTANT NOTE: *These solutions are further illustrated in the Installation and Licensing Options technical white paper.*

Installation Options

Two Directories

If you only have two directories to synchronize, one copy of UnitySync with connections to both directories works best. This implementation solution is perfect for:

- two companies that have merged
- two interdependent organizations

To install UnitySync on any server in your network you'll need:

- a license for one base copy of UnitySync
- a directory license for each directory being synchronized
- a license for an optional lab copy for testing

Multiple Directories - Centralized

If there are multiple directories controlled by a single management entity, then synchronizing with a master/hub directory is a better option. We call this a master/hub-spoke implementation.

This implementation option is perfect for:

- multiple interdependent organizations
- a parent organization with three or more sub-organizations
- a parent company with three or more autonomous subsidiaries
- an umbrella organization with multiple interdependent parts/sub-organizations

To install UnitySync for a master/hub-spoke configuration you'll need:

- a license for one base copy of UnitySync
- a directory license for each directory being synchronized
- a license for an optional lab copy if you do a lot of testing

Multiple Directories - Decentralized

If a regular master/hub ? spoke directory implementation isn't the best solution, for various reasons, there is yet another implementation option. Installing UnitySync on or near remote directories gives admin control to each site/directory location.

This implementation option is perfect for:

- low bandwidth or irregular connectivity situations
- when administration at each directory site is required or preferred

To install UnitySync this way, you'll need:

- a license for the initial copy of UnitySync

- a license for each additional copy of UnitySync, per server
- a directory license for each directory being synchronized
- a license for an optional lab copy if you do a lot of testing

Questions to Consider

Once you've chosen an implementation option, it's time to think about things like:

What server(s) are your Source and Destination directories on?

Each server has unique identity and authentication information. Know which servers you are using and be prepared with the IP, login and password. You'll need this to configure the UnitySync connections in just a bit.

Which objects will I want to synchronize?

UnitySync synchronizes all user objects in the Source directory to the Destination directory by default. To change that, customize Object Types, the Source Context, and Optional LDAP Query Filters to narrow the Source query definition.

You may also utilize the Exclude function to narrow your scope further.

Where do I want objects to go?

UnitySync can create a sync container on the root of the Destination directory. To change the location, add a Placement DN as well as the Structure Name or simply point it to an existing container on the Destination directory. Just changing one, or both, will send all objects to wherever you specify and put them in a flat directory. To replicate some or all of the Source directory's structure, adjust the Sync Level parameter.

What sort of logs and e-mail notifications do I need?

UnitySync automatically writes a log file with a Log File level of 2-Lowest for each sync process. Low level log files will keep track of all modifications, additions, and deletions processed in the synchronization run. Consider using a Log File level of 3-Detailed for more information while troubleshooting. Levels 4 and 5 contain a lot of information and aren't often necessary unless specifically requested by Directory Wizards Technical Support.

UnitySync can also send email notifications with the log files to anyone that you choose. Set email notification at the global level to receive notifications for all synchronization processes, or set it only for a specific connection.

How do I want format display names?

To choose how you would like users to see display names in the Global Address List (GAL), consider options on the Display Name tab. It can be set to show the last name first, the first name first, or even add a label after the display name, providing an identifier for synced contacts.

Will I need any customization?

UnitySync comes configured with recommended default settings for every directory type but, sometimes, the needs of your environment will require customization. We can help whether you need to sync an attribute unique to your environment, alter default mappings to produce a required result, or just about anything else under the sun. We recommend you contact Directory Wizards Technical Support at support@dirwiz.com for guidance with needed customizations.

Pre-Synchronization Checklist

We recommend you make sure you have all of the following decisions made prior to attempting to run your first synchronization:

- Source and Destination directory IPs, logins, and passwords
- synchronization objects defined
- email notification settings
- user (administrator) access levels decided
- display name format chosen
- customizations discussed with Directory Wizards Technical Support who will help you implement any changes to the defaults

Prepare Directory Servers for Use With UnitySync

UnitySync can be used with an ever-growing list of different directory types! As of this writing, we can support the following directory types:

- Active Directory
- Active Directory Lightweight Directory Services
- Amazon Simple AD
- Azure
- CSV files
- Google Workspace (formerly GSuite)
- LDIF files
- Office 365 (o365)
- ODBC link to databases
- OpenLDAP
- Sunone
- ViewDS

At minimum, you will need a service account to connect with both your Source and Destination directories. Please click on the directory type for more information on requirements for the specific directory type.

Prepare Directory Servers: Active Directory (with or without Exchange)

The smallest element of Active Directory (AD) is the domain. Each domain is controlled by a domain controller. This domain controller stores user account information, permissions and some basic contact information for that domain. A domain is always part of only one greater forest. A forest is comprised of one or more domains. Within the forest, domains automatically set up trust relationships so they can assign permissions between the domains.

Domains (individual domain controllers, or DC's) do not store information from other domains. However, each domain maintains a least one complete database of information from all the forest's domains. This is called the global catalog server (GC).

Both a domain controller and a global catalog server are separate LDAP servers:

- Domain Controller: port 389 (SSL 636)
- Global Catalog: port 3268 (SSL 3269)

If you would like to read only one domain's worth of information you can connect directly to that DC. If you need to read from many domains (whole forest) you'll need to read from a GC (a DC that contains the GC).

The GC ports are read only. UnitySync can only write to DC ports 389 or 636.

Active Directory with Exchange

Exchange is a mail server. It uses Active Directory to store its configuration data, account information etc. When Exchange is loaded on AD, it modifies the LDAP schema of AD to add Exchange specific attributes. When syncing what you are really doing is reading/writing to the underlying AD DC's. When writing to an AD Destination (and this includes one with Exchange loaded), you must still specify the IP address of a DC, not the Exchange Server itself.

When creating a new UnitySync connection, the Destination Map Template you select determines if the AD objects created will include Exchange attributes (i.e., be mail enabled). If you want your UnitySync connection to create Exchange mail enabled objects, choose a Destination map template that includes *mail enabled* in the template name.

Configuring Active Directory Login ID

Setting up a UnitySync Account

You will need to create an account that will be used by our programs to read and/or write to your directory. It is preferred that your UnitySync credentials be granted domain admin privileges; if you are not able to provide domain admin, apply Special Permissions as outlined in this knowledge base article. Additionally, you may encounter a

few other issues if you are unable to use a domain admin account. See the article which outlines these potential concerns.

- Launch Active Directory Users and Computers (AD U&C).
- Open the tree until you find the appropriate container.
- Right-click on OU and select New/User.
- Enter the appropriate account information.
- Press Next
- You may use whatever password you like. We recommend checking "User cannot change password" and "Password never expires" to eliminate the need to maintain this account.
- Select Next, then Finish.
- You should now see the account you created in the appropriate OU.
- To add your new user to the Domain Admins group, double-click on the Domain Admin group in the Users container.
- Select Members, Look In: Entire Directory. Select your new user account and press Add.

User ID Syntax

Active Directory uses a 'Domain Component' structure for its user ID. When you setup Active Directory you assigned an internet domain name to it (i.e. dirwiz.com). An example of a user ID would be: UnitySync@dirwiz.com

Prepare Directory Servers: Amazon Simple AD (a component of AWS Directory Service)

As of UnitySync v2.5, Discovery and Sync of Amazon Simple AD directory is supported.

Simple AD, a component of Amazon Web Service (AWS), is a Samba-based directory in the cloud which (among other things) contains your Users and Groups. UnitySync is able to Discover (read) Users and Groups from your Simple AD directory. UnitySync is able to write to Simple AD creating contacts.

UnitySync is able to Discover (read) Users and Groups from your Simple AD directory. Data discovered from Simple AD can be written to any Destination LDAP directory or Office 365 (O365) directory, exported to CSV or written to almost any database.

UnitySync is also able to Sync (write) to Simple AD creating contacts. You can discover any supported source type (LDAP directory/CSV/ODBC) and write to your Simple AD, creating contacts.

Syncing with AWS Simple AD can use the same templates as Active Directory in v2.4. As of v2.5, templates specific AWS Simple AD are available.

IMPORTANT NOTE: *If using a version of UnitySync previous to v2.5, be sure to select active-dir templates that are NOT* mail enabled. Using active-dir templates has not been tested in versions prior to v2.4.**

Login ID Syntax

You will need to use the default administrator login in order to read from or write to AWS Simple AD.

UnitySync Office 365 login requirements

If you are running any version prior to v4.6, you must upgrade to the latest version to sync with o365.

Connections may be configured to read your O365 directory as a **Source**, discovering Users, Contacts and/or Groups. These objects may be synced to create contacts in any other supported Destination directory type (Active Directory, O365, etc).

Likewise, connections may be configured sync to your O365 directory as a **Destination**. When syncing to an O365 Destination, contacts will be created.

1. System Requirements

Ensure your UnitySync server meets the Special Requirements for o365 connections.

2. Required Login information

Source or Dest tab ID:

The specified o365 User account must be:

- **** A licensed o365 account**** (Administrator or other custom account created for UnitySync).
- **Excluded** from any policy requiring Multi Factor Authentication (MFA)
- **Enabled.**
- **Password NOT expired.**

ID name format: Use the same login format you would use when logging into O365 on line:

i.e. AdminAccountName@YourDomain.onmicrosoft.com

Password: The password that corresponds to the login ID specified

3. The User must have Permissions to read/write an O365 tenant

We highly recommend use of an **O365 Admin account**. Microsoft does not make it easy to create a non Admin account with the necessary access for UnitySync to perform the required powershell commands when writing to O365.

Discovery of O365

We highly recommend use of an O365 Admin account. Alternatively, you may assign minimum read access to your UnitySync O356 login ID to be used for O365 Discovery.

- For example, create an unlicensed Office 365 user account *without* O365 admin rights.
- For *view only* access to O365, add user to the "View-Only Organization Management" admin role in the Exchange Admin Center. This role should provide UnitySync the rights needed to run the powershell commandlets utilized by UnitySync Discovery.

Discovery, commandlets required:

Get-EXORecipient
Get-DistributionGroupMember

Syncing to O365:

We highly recommend use of an O365 Admin account. Microsoft does not make it easy to create a non Admin account with the necessary access for UnitySync to perform the required powershell commands when writing to O365.

That said, it is possible for a Non Admin account to sync to O365.

Sync, commandlets required:

The following commands are utilized by a UnitySync Sync process:

Remove-DistributionGroup
Remove-MailContact
New-DistributionGroup
New-Contact New-MailContact Set-Group
Set-DistributionGroup
Update-DistributionGroupMember
Set-Contact
Set-MailContact

Additionally, here is an Sample O365 RBAC script to reduce permissions.

IMPORTANT NOTE: This script was provided by a client as a sample script which allowed them to create a non Admin account for UnitySync to sync to o365. Your script may be different depending on your preference and environment. Using the script as an example, you can try to create a custom account with minimal access.

For more detailed information about setting read/write permissions on O365 User accounts, refer to Microsoft tech articles and/or reach out to Microsoft technical support:

Refer to: TechNet Overview of Built-in role groups

Refer to: TechNet View-only Organization Management

Refer to: Permissions in Exchange Online

Refer to: Create an unscoped role

For more information about O365 syncs, please refer to the **O365 KB articles** and the **UnitySync Administrator's Guide**.

Prepare Directory Servers: Active Directory Lightweight Directory Services (AD LDS)

Active Directory Lightweight Directory Services (AD LDS, and formerly known as ADAM), is so customizable that it's difficult to create a generic default configuration for it. That said, here are configuration tips for syncing to/from AD LDS. If you have trouble configuring or running test connections to your AD LDS directory, contact Support@dirwiz.com for assistance.

Configuration Requirements

- You can use the default AD LDS schema
- Create a dedicated UnitySync user account.
- If you need to synchronize an expanded attribute set in AD LDS, you will need to create a custom sourcedef and custom mapping. Please contact support@dirwiz.com for guidance.
- The UnitySync AD LDS User Account may be made a member of the AD LDS admin group or you may assign special permissions to grant the UnitySync user account permissions to individual Destination AD LDS containers. Details on how to configure the AD LDS User Account follow.

Create the UnitySync AD LDS User Account and Sync Container

Find below the steps required to create the UnitySync AD LDS User Account and the AD LDS sync container. Perform the following steps in ADSI Edit:

- Create the Sync container (where the sync will create objects):
 - Select the root in which to create the new container.
 - Click New > Object > Organizational Unit > Next
 - Enter the container name value (i.e. Jons World), click Next, click Finish.
 - If your AD LDS user login will not be an Admin account, you'll need to apply Special Permissions , giving your non Admin user account write access.
- Create the AD LDS UnitySync user login account
 - Select the root in which to create the UnitySync user account.
 - Click New > Object > User > Next
 - Enter the user name value (i.e. JonDoe), click Next, click Finish.
 - Right click the new user (i.e. CN=JonDoe), select Reset Password.
 - Enter the password, confirm password and click OK.
 - Open CN=Roles.
 - Right Click CN=Readers, select Properties.
 - Double click the attribute 'Member'.
 - Click Add AD LDS Account.
 - Enter the DN of your UnitySync User Account (i.e. cn=JonDoe,dc=acme), click OK.
 - Click OK
 - Right click your UnitySync user account (i.e. CN=JonDoe), select Properties.
 - Double click attribute msDS-UserAccountDisabled, select FALSE, click OK.

- Double click attribute msDS-UserDontExpirePassword, select TRUE, click OK.

IMPORTANT NOTE: The *DontExpirePassword* setting is not required, but recommended. The password is specified in each UnitySync connection. If this is set to False and your password will expire and be reset regularly, you must modify your UnitySync connections to include the new password at the same time.

Login ID Syntax and Sync Container Configuration

Preferred login ID syntax is user@domain.com. Please be sure userprincipalname is set to this format.

In order for your UnitySync AD LDS User account to have appropriate permissions to manage objects in your AD LDS directory, you may make your UnitySync AD LDS user account a member of the ADMIN group (recommended).

OR

You may apply 'Special Permissions' on the Sync container, granting the UnitySync AD LDS user account permissions to just that one container.

To Configure Special Permissions

You must first complete the steps outlined above for Creating your ADAM UnitySync User Account and Sync Container.

Refer to instructions on configuring your LDS user permissions in the related Microsoft Technet articles: <https://technet.microsoft.com/en-us/library/hh831593.aspx>

IMPORTANT NOTE: Please keep in mind when configuring the AD LDS User permissions, the UnitySync user must have permission to create, modify, and delete child objects (including contacts and sub-ou's) within the Sync Container.

Tips for Configuring Your Connection

- Select the Map Template for adam.
- For AD LDS Destination connections:
 - Destination tabs Placement DN is required.
 - On Sync, *Inetorgperson* object types are created.
- For AD LDS Source connections:
 - Source tabs Selection DN is required.
 - On Discovery, *Users* and *Inetorgperson* types are pulled.

Prepare Directory Servers: Microsoft Azure

UnitySync supports Discovery (read) of Microsoft Azure User objects. Refer to our Azure KBs for some helpful information.

To set up Azure for use via LDAP, please visit Microsoft's information to Configure secure LDAP (LDAPS) for an Azure AD Domain Services managed domain.

Data discovered from Azure can be written to any Destination LDAP directory (including Active Directory) or Office 365 directory, exported to CSV or written to almost any database.

Login ID Syntax

You will need to use the default admin login in order to read from Azure.

Prepare Directory Servers: Domino/Lotus Notes

Configuration Requirements

For Notes, you'll need at least a Notes R5 server for directory synchronization. You'll also need an account that has add and delete access to the names.nsf database on the Notes server. This will sync objects to the primary notes addressbook. To create objects in a secondary notes addressbook, see the alternate instructions below.

Configure LDAP for UnitySync

1. Open the server's address book as an administrator using a Notes client.
2. Select > Server/Servers and open the properties of the server you plan to sync with
3. Enable LDAP
4. Ensure that the TCP/IP port is valid and does not conflict with other software
5. Select Ports > Internet Ports > Directory from the server properties screen.
6. Set Name & Password authentication to Yes
7. Save the server configuration screen
8. Select Server > Configurations
9. If there is not a configuration entry for your server click Add Configuration. Otherwise edit what's already in place.
10. If this is a new configuration we recommend using a global configuration by setting Yes to "Use these settings as the default settings for all servers". Once this is checked an LDAP tab will appear.
11. Select the LDAP tab
12. Set "Allow LDAP users write access" to Yes. (the defaults are OK for the other options)

IMPORTANT NOTE: *If you have just enabled LDAP you may have to restart the server or consult the Lotus Notes documentation on starting the LDAP service.*

Login ID Syntax

You must use the full Lotus Notes User ID when authenticating with UnitySync. UnitySync uses LDAP to access Lotus Notes. Below are some examples of translated User IDs:

= Notes	= LDAP
Eric Nichols	cn=Eric Nichols
Eric Nichols/Dirwiz	cn=Eric Nichols,o=Dirwiz
Eric Nichols/Support/Dirwiz	cn=Eric Nichols,ou=Support,o=Dirwiz
Eric Nichols/Development/Support/Dirwiz	cn=Eric Nichols,ou=development,ou=Support,o=Dirwiz
Eric Nichols/Dirwiz/US	cn=Eric Nichols,o=Dirwiz,c=us

Using Directory Assistance to Write to a Secondary Addressbook

Start with our knowledge base article [Can UnitySync read/write secondary addressbooks?](#) for an overview. Then visit our technical white paper, [Notes Secondary Addressbook](#), which specifically details the steps to configure your Secondary Addressbook properly.

Once you have finished configuration, if you have trouble syncing with your secondary addressbook, please refer to this knowledge base article: [Configured sync to write to Secondary Addressbook, but still getting errors](#).

That's all the configuring that's needed for UnitySync to automatically read records from all addressbooks. If necessary, use excludes options to limit the objects that are actually pulled from the Source addressbooks.

Prepare Directory Servers: Google Workspace (formerly G Suite)

As of UnitySync v2.9, Discovery (read) and Synchronization (write) of Google Workspace is supported. This version, and newer versions, of UnitySync can Discover Users and Contacts from Google Workspace, and Sync Contacts to Google Workspace.

Google to AD example:

In a UnitySync connection with **Google Workspace as the source**, the Discovery phase runs to obtain the latest Google Workspace data. The **Sync phase** of that connection will use the data, writing to almost **any** Destination LDAP directory (including Active Directory). The data data can also be synced to an o365 tenant, be exported to a CSV file, or be written to almost any database (via ODBC).

AD to Google example:

In a UnitySync connection with **Google Workspace as the destination**, data will be discovered from the directory specified on the source tab. Then **on Sync**, the data will be written to the destination Google workspace.

IMPORTANT NOTE regarding Syncs INTO Google: On Sync, a Mod Object (modification of an existing Google contact) will result in a Delete/re-Add of that Google contact.*

Login ID Syntax

Authenticating with the Google workspace (source or destination) requires the use of the **default admin login** in order to access the Google Workspace data. Using that login, there is a one time verification check in order to receive an Access Token.

Prepare Directory Servers: All Others

UnitySync supports Discovery (read) and Synchronization (write) to many directory types. If preparing your directory type is not covered in a specific topic, please reach out to our Technical Support Team if you have any questions regarding preparing your directory server for use with UnitySync.

The UnitySync User Interface (UI)

The UnitySync Manager User Interface (UI) gives you the power to manage all synchronization processes in one place. Manage individual connections, log files, global settings, and access help files all from the UnitySync UI.

Launch Page

Upon first installing UnitySync, or when you have not selected a connection you will see the Launch Page.

The first line of the UI, the TopRow, contains global functions affecting all of your individual connections. You will see the current Connection name (or a blank drop down list if no connection is selected), and buttons to create a New connection, configure your Global SMTP settings, view the status of processes inSync Monitor and Doc to find help when you need it. Please see the following section for more information on these TopRow global functions.

The Launch Page also includes version information, your Serial Number, and License Key entry and information. As of v4.0, you may request a new key here using the Request Key button, which uses mailto functionality to provide all relevant information needed in the body of an email to keys@dirwiz.com](mailto:keys@dirwiz.com).

Once you receive your key, enter it in the License Key box and click the Save Key button.

In the upper left is the drop down selection list of existing connections. Here you may select an existing connection to edit/run. Once you select the connection, you will see multiple tabs and parameters to configure. Refer to Configure UnitySync sections for details on each section/tab of the UnitySync UI.

TopRow Global Settings

Configuration items on this row are global and may affect all connections. You will see the TopRow above tabs no matter which connection or tab is active.

Connections Drop-down

Select any configured connections from this drop-down list and it will become the active connection. If you haven't configured a connection yet, move on to the next section to find out how.

New

Click here to create a new connection. The New Connection window will load.

In the Connection Name box, type a name for your connection. Then, select the appropriate Source and Destination Map Templates for your directory types. The Sync Engine will automatically change depending on the Map Template type, but this can also be manually adjusted if you so desire.

Click Save to preserve your new connection.

Global SMTP

Select this to define global server and reply addresses for error and/or summary email alerts.

In the SMTP Server Settings, indicate your SMTP server and Reply Address. Then provide the email address to receive summaries and error alerts. You can also choose the urgency of the message and whether or not to send logs with each type of email. Use the test buttons to insure you will receive any emails generated by UnitySync.

More information on Global SMTP is covered in the information on the Email Notify tab.

Sync Monitor

Selecting Sync Monitor will display a separate window with a process list of all running connections.

If any connection is running, you will see the detail under the heading shown in the window.

- Connection - connection name
- Program - the executable currently running (ldif-ldap, csv-ldif, etc.)
- Mode - what process is running (disc, sync, sim, or unsync)

- PID - process ID
- Time - time elapsed since process started

UnitySync processes information so efficiently, however, you may not see any processes running at all! If this is the case, the process likely finished before you had the opportunity to open the window.

Doc

This button provides you quick access to this Administrator's Guide from the UI. Click it for quick answers to your questions. The Administrator's Guide will open in a new browser tab.

Configure UnitySync

General Tab

Configure batch run actions, set logging levels, unsync, forcemods, configure LDIF threshold and optional LDIF Source file location.

Source Tab

Configure Source specific options. The configuration options of this tab will differ depending on the Map Template and Sync Engine.

Destination Tab

Configure Destination specific options. The configuration options of this tab will differ depending on the Map Template and Sync Engine.

Display Name Tab

Define the Display Name Format of synced objects.

SMTP Address Tab

Options to manipulate the SMTP address of synced objects.

Custom Tab

Custom configuration files may be defined for each individual connection. Object (attribute) Mapping, List (attribute) Mapping, Source Definition, Destination Structure, Lookup Tables, Exclusions, Raw Configuration, and more.

Email Notify

Email notification of connection run results may be configured globally or per connection.

General Tab

The General Tab provides information about your connection and allows you to select your general configuration options.

Connection Specific Information

Map Templates

The core of the UnitySync process is attribute mapping between your Source and Destination directories. The Map Templates, which you selected when you created your connection, identify the Source and Destination directory types this connection will sync between.

Sync Engine

Sync engine is selected when you create your connection. The Sync Engine identifies the directory formats this connection will sync between. Most syncs will use the default LDAP Source/Dest engines.

Connection Box

Options regarding your currently active connection, including:

- Copy - copy your current connection
- Rename - rename your current connection
- Delete - delete your current connection

Description Box

This is a free form text field in which you may enter informational details regarding this connection.

Action Box

The items in this section are used to execute your desired function against the current connection. Each sync is made up of two primary phases: Discovery (read) and Sync (write). Additionally, you may elect to include a Simulation phase. The connection can be configured to run one, two or all three phases, and other options in this section are helpful for your connection runs.

When you execute any of these actions from the UnitySync UI, a console window will pop-up showing you the progress. You can close this window at any time by using the X in the upper right corner. If you then wish to view the details of the last connection run, you can access it using the View Console button (details below).

Actions include:

Batch Run

Batch Run will execute any of the checked (enabled) processes below the Batch Run button in the UI. You can use Batch Run by clicking the button or via command line options.

View Console

View Console will bring up the Console window from the last process run.

Discovery

When enabled, the Discovery Phase performs a read of the Source data and outputs an LDIF file, ldif.txt. This LDIF file will be used by the Simulation and Sync phases. You may also configure other connections to use this LDIF file as input. In order to process the latest Source data each time, Discovery should always be enabled. Check the box to select for Batch Run or click to run Discovery only.

Simulate

When enabled, the Simulation Phase performs a simulated write to the Destination. The Simulation output will show proposed Adds/Mods/Deletes. This and other details are written to a Simulation log file. When Simulation is enabled, Simulation Tolerance may be set. If a Tolerance Violation, or any other Error is logged during Simulation, the Sync phase (if enabled) will not be executed. The only exception is Special Excludes. Check the box to select for Batch Run or click to run Simulation only.

Synchronize

When enabled, the Sync Phase will read the Idif.txt and perform adds, deletes and modifies to the Destination directory. Check the box to select for Batch Run or click to run Sync only.

IMPORTANT NOTE: *If both the Simulation and Sync phase are enabled, then both will be executed on Batch Run. Any time the Sync phase is enabled, the final write phase will run, which may apply Adds/Deletes/Modifies to your Destination directory. These Run settings apply to both a manual run from the UI as well as a command line (automated script) execution.*

Last Successful Run

The date and run time of the last Discovery, Simulation and Sync is displayed below this heading.

IMPORTANT NOTE: *Refer to Automating (Scheduling) your UnitySync connections for additional information on scheduling Sync runs.*

General Tab: Unsync

The UnSync function on the General tab will remove the synced objects from your Destination directory.

IMPORTANT NOTE: *Please be very careful using Unsync! Using Unsync will **NOT** simply undo your last action. It will remove **ALL** objects that were created using that connection.*

Unsync is a handy, but powerful, tool that deletes synchronized objects and structures from the Destination directory. Keep in mind that each UnitySync connection has its own Unsync. So, one Unsync will only remove objects that were synchronized with a specific UnitySync connection. Run an Unsync for each connection to get all objects and structures. Note also that Unsync will remove **ALL** synced objects and structure and **NOT** just the results of your last synchronization run. Think carefully about the possible ramifications of using Unsync before pushing the button!

Using Unsync is a great way to test the configuration of your first connection(s). If the results of the synchronization run are not what you would like them to be, run an Unsync. Next make the necessary configuration changes and then try another synchronization run. Running an Unsync/sync during the initial setup and testing of a connection helps avoid any configuration problems that certain configuration changes between synchronization runs might cause. Use Unsync carefully and thoughtfully once synchronizations are running in a production environment.

IMPORTANT NOTE: *Unsync will remove everything within the scope of the sync from the Destination. It will not simply undo the last synchronization actions. If you are unsure if Unsync is the best option for your situation, please contact Technical Support for assistance. Often, we can help you correct issues without having to delete all Destination objects and start again.*

To run an Unsync:

1. open your Connection
2. select General tab
3. click on Unsync

A pop-up window will tell you how many objects will be deleted upon Unsync and ask, 'Are you sure?' that you want to delete the objects. If you are, click Yes. If you aren't, you can Cancel out now to stop the Unsync.

Finally, note that Unsync will remove the objects it deletes from any groups on the Destination. If Unsync is performed in error, the most expedient way to return the Destination objects in the state they were prior to the Unsync is to restore them from a backup. Simply re-syncing the connection will not restore group membership, and may result in different attribute values, particularly those that are considered no mod attribs.

General Tab: Forcemods

In the event that your data gets out of sync through corruption or user manipulation, use the following steps to force the sync to zero out the database information and reapply all synced data. This will add missing objects, modify all objects and perform outstanding deletes. (If you **do not** want to perform the outstanding deletes, run the Recovery procedure instead).

- Select your Connection
- Click the Force Mods button located in the Action section of the General tab.

This function will zero out the internal database entry for each synced object, and run a Sync to rebuild the database information. **Forcemods will add missing objects, process outstanding deletes and apply a MODIFY to all previously synced contacts.**

This function may be automated via the shell.exe execution of your connection during your scheduled sync runs.

IMPORTANT NOTE: *It is not recommended you automate forcemods on ALL scheduled sync runs. Force Mods causes **every** object to be modified on **every** sync run. This is generally unnecessary and to time consuming to run on a regular basis. If you know you have regular modification/deletion of destination contacts, it might be recommended to automate your Forcemods connection once per week or month.*

If you continue to have data corruption issues after performing Forcemods, it may be necessary to run the Sync Recovery Procedure.

General Tab: Logs Box

This section allows you to select your desired logging level for Console, Log File and Event Log, as well as view log files.

The default setting for Console logging is 2-Lowest. Used for troubleshooting purposes, Console logging can only be viewed when a connection is executed from the command line. The default level of detail is recommended for normal use, but if you would like to see greater detail to confirm desired results, 3-Detailed is a good option.

The default setting for Log File is 2 - Lowest. Log Files show the results of each action you've chosen - Discovery, Simulation, or Synchronization - and are available to download and view in a drop-down in the Logs section after each run. The default level of detail is recommended for normal use, but if you would like to see greater detail to confirm desired results, 3-Detailed is a good option.

IMPORTANT NOTE: *It is recommended that you leave the Log Files setting at 2-Lowest on a regular basis. Although 3-Detailed gives the best level of detail for troubleshooting, the log files are significantly larger. If you do not have log file management processes in place, larger log files can fill up disk space and result in a low disk space condition. To prevent this, we recommend Log File level set to 2-Lowest regularly, log file maintenance, and configuring your Operating System to alert you to low disk space conditions prior to them becoming critical.*

The default level for Event Log is 3-Detailed. This setting allows for errors to be logged in your Event Viewer application logging. The default level of detail is recommend for normal use, but you can determine which events you would like to be logged:

1. Disabled
2. Lowest (Only Fatal Errors that halt a sync)
3. Detailed (Return 1 & Soft error, mod error, etc.)
4. Verbose (All, including No Error runs)

Download Log

Click the button next to the drop-down list to download a specific log directly from the UnitySync UI. The most recent log file will be at the top of the list and therefore visible in the UI by default, but you can also select any existing log from the drop-down to open. Your log will open in your default text editor.

General Tab: Logging Orphans (and deleting orphans)

This article discusses the features to identify and delete orphans using your UnitySync v3.x connection.

This article also pertains to v2.x in so far as the method to enable and review output from **Log Orphans**. However, the resulting orphans.txt format is different and the delete orphans feature itself is not available in v2.x.

What is an Orphan?

Occasionally, your Destination sync container may contain contacts that no longer appear on the Source. We call these objects "orphans". An orphan is any object that exists in the Destination sync container but is not within the scope of the discovery of the Source.

Most often, these are objects that were previously synced by UnitySync, then deleted on the Source but the deletion did not also occur on the Destination. *NOTE: The orphan will be deleted, even if it wasn't created by UnitySync.*

Sometimes, orphans occur when you have opted for *Delete Processing: Ignore Deletes* on the Destination tab. (Even when you choose to Process Deletes on the Destination tab, a delete of an object is only attempted once. If the delete cannot occur, for whatever reason, when it is attempted, it will not be attempted again).

As of UnitySync v2.0, if you select the General tab option to * 'Log orphans after Synchronization'* The Sync phase will identify orphans for you, producing a file that contains the DN's of all orphaned objects. A list of orphans is also included just above the Run Summary in your sync log.

How do I enable Log Orphans on Sync?

1. In your connection, on the General tab, check the box to *Log Orphans on Sync* and Save.
2. Run a Discovery and Sync on the connection. The sync will run as usual, orphans will be listed in the sync log, AND a file is generated: `\Connections\YourConnectionName\orphans.txt`
3. Review the orphan.txt file. The full DN for each orphan is on single line.

What can I do with Orphans.txt?

1. The full DN for each orphan is on single line.
2. **To manually delete the orphans**, review Orphans.txt, determine if all, or only some, are appropriate to delete. Then you can **manually** delete the appropriate objects from the destination.
3. **Alternatively to manually deleting the orphans**, there is a feature allowing UnitySync to **batch delete** ALL DN's in the orphans.txt file. While this delete orphans feature is available in v3.0, the instructions remain undocumented. (Undocumented due to the fact it allows a connection to delete destination objects *not currently controlled by the connection*).

So please contact Technical Support Team and we can provide assistance with the delete orphans feature.

IMPORTANT NOTE: *The information in ophans.txt is the comparison of database files of the current connection to all the objects that exist in the destination Sync container. This means, **if you have manually created contacts, OR more than one connection syncs to the same Destination Sync Container, each connection's Log Orphans function will identify the objects that belong to the other connection as orphans.** This can be avoided by following our best practices guidance on creating connections which is that each connection should point to a unique Sync OU. And do not manually create objects within the Sync OU.*

Note About v2.0 usage of Log Orphans The in v2.x, If you use orphans.txt and ldifde (or other LDAP utilities) to process the deletes with UnitySync, the orphans.txt generated by v3.0 is NOT compatible with your ldifde solution. Please contact our Technical Support Team for guidance.

Recommendation: We do **not** recommend that you enable orphan logging on all syncs. Running orphan check on every sync will slow down your connection. Instead, only run the orphan check when you know or suspect you have orphaned objects on your Destination that you want to identify and/or delete.

Ref:3213

General Tab: More Options

Simulation Tolerance (Optional)

The optional Simulation Tolerance settings only apply when Simulation is enabled. If any of these tolerance levels are set, the Simulation will determine the number of Adds, Mods and/or Deletes that will occur at sync time. If the Simulation determines the number of Adds, Mods or Deletes exceeds the tolerance level specified, the Simulation stage will end with an Error condition and the Sync stage will not take place.

LDIF (Optional)

Read Location: Instead of using this connection to run Discovery, you may select an ldif.txt from another location; usually another connection directory. When a Read Location is specified here, the Discovery Option for that connection fades to indicate it is disabled.

Encryption Password: If your Source Sync Engine is LDIF, meaning your Source LDIF file was created by an alternate connection, the Source LDIF may be encrypted. If so, then you must enter the same encryption password here so that the LDIF file may be decrypted. Generally, encryption is only used when the connection that created the LDIF file is running on a different UnitySync system and the LDIF file must be transported (i.e. ftp or email) from the originating UnitySync server to the local UnitySync server.

Source Tab

The parameters available on the Source tab will differ depending on the Source Directory type. Configuration options available for each directory type are outlined in separate sections as follows.

IMPORTANT NOTE: Any time you are making changes to parameters that may affect the scope of the Source (like those on the Source tab) it is very important to disable Synchronization and enable only Discovery and Simulation. Run the connection to confirm desired results in both of these logs before enabling the Synchronization phase.

Source Tab Configuration: Standard LDAP Directories

LDAP is the protocol required to access any LDAP directory. LDAP directories include Active Directory (with or without Exchange), AD LDS, Notes, etc. The following configuration items may be available on the UnitySync Source Tab for a standard LDAP directory. If an LDAP directory does not support a particular function, the related UnitySync option will not be available as a configurable option on the tab.

LDAP Connection Information Box

IP: IP address or host name of the Source system. For an Active Directory Source, if you want to pull the entire forest, specify the IP address where the Active Directory Global Catalog (GC) resides. Otherwise, you must specify the address of a Domain Controller (DC). In v4.0 and later you may specify more than one IP or hostname separated by commas to act as failover in case the first IP/hostname cannot connect.

IMPORTANT NOTE: If relevant connections must use the same values/order. Otherwise, you may receive a license error as UnitySync will count all the first listed IP's as separate directories. Say you are using 192.165.51.111 and 192.165.51.10. Hostnames for these servers are ServerA and ServerB. Each time you want to use list B as a failover for A, you want to be sure to list it the same way each time, and not use A as a failover for B. You can use either hostname or IP but please be consistent.

Encryption: The SSL parameter can be used to set the port and may enable additional functionality.

- **No Encryption:** (default) sets standard port (389), no encryption
- **StartTLS:** allows encryption over standard non-encrypted port (389)
- **SSL/TLS - Basic:** sets SSL standard port (636), enables encryption but not expiration checking of certificates.
- **SSL/TLS - Expiration Check:** sets SSL port (636), enables encryption and expiration checking of certificates.

IMPORTANT NOTE: *Directory Wizards strongly recommends using SSL/TLS options if you desire encryption, but provides the StartTLS option as an alternate. If you have any questions regarding the security of your connection, please contact our Technical Support Team.

ID: The DN of an administrative or authenticated account. This is required information when using Active Directory and Office 365. It can be left blank for anonymous login for other LDAP directory types (if enabled). When restricting the Source directory to read-only access this account does not have to be administrative.

Password: The password that corresponds to the ID specified.

TEST button: Clicking this will test that the information provided in the other fields is valid to provide access to the Source location.

Object Types Box

Available Object Types will differ depending on which directory type you are pulling from. Examples of Object Types that might be available are: Users, Contacts, Groups, Folders, Hidden. Only those Object types selected here will be included in the sync.

HINT: When syncing from an AD/Ex20xx Source, if you wish to only pull User Accounts with a Mailbox, see our knowledge base article for detailed information.

Optional - Source Context Box

By default, the sync will read from the root of the Source directory. You may choose to pull from a specific location by entering the complete DN syntax of the Source container here. For example, to pull a specific container in AD, the format is:

```
ou=Contacts,ou=London Office,dc=domain,dc=com
```

IMPORTANT NOTE: For AD LDS, OpenLDAP and others, a Source Context Selection DN is required.

Selection DN Syntax

When pulling objects from the Source system, the default (blank Selection DN) will start at the top of the tree. You may override the default by specifying a Selection DN in the Source Context field to indicate where in the LDAP tree you would like to start. This applies if you want to pull only one small piece of the tree (container/ou) rather than pulling the whole tree. Multiple selections may be specified separated by a | symbol.

Example Active Directory syntax:

```
ou=MyOU Name,ou=MyParent OU Name,dc=domain,dc=com
```

HINT: Use ADSI to see complete LDAP syntax for your directory.

Example other LDAP directory syntax:

```
ou=MyOU Name,ou=MyParent OU Name,o=Top
```

IMPORTANT NOTE: For the Source systems of ActiveDir, Exchange, Netscape, NDS, & Notes the starting point is automatically detected. If you leave this option BLANK you will pull the entire Source directory. This is the default. AD LDS, OpenLDAP and some other Source types may require a Selection DN be specified to identify the root to pull from.

Optional - LDAP Query Filter Box

When the Source is an LDAP directory, UnitySync sends an LDAP query and asks for specific information. Filters can be used to include and/or exclude objects based on their values. This option is only valid against an LDAP Directory Source (not odbc, ldif, csv etc). See Filters: Optional - LDAP Query Filter for more information.

Source Tab Configuration: Office 365 (O365)

Options for Office 365 (O365) are a simplified version of standard LDAP connections. Please note that O365 templates are only available in the Windows version of UnitySync.

O365 Connection Information Box

ID: The login of the default administrative account. This is required information when using O365. Please see more detailed information in the O365 Overview article in our knowledge base.

Password: The password that corresponds to the login ID specified.

Object Types: Available Object Types for O365 Sources include User Mailboxes, Mail contacts, Dist Groups, and Hidden. Only those object types selected here will be included in the sync.

IMPORTANT NOTE: In software versions 2.8 and earlier, Hidden objects were discovered by default OR you had to filter them out using the HiddenFromAddressListEnabled attribute. If you are using a software version of 2.8 or earlier, you will not see Hidden as an option here, and your Discovery will pull Hidden objects. If you are upgrading, please pay special attention to the upgrade instructions in the Release Notes and Installation Guide for your new version. If you need assistance with a filter or the upgrade procedure, please reach out to our Technical Support team.

Optional - Filter: As with other standard LDAP directory types, you are able to filter what you discover from O365. Filters can be used to include and/or exclude objects based on their values. O365 filter queries utilize powershell, but filter syntax has been crafted to be very similar to LDAP filter format.

For in depth information about using the Filter functionality for O365, you will need to know your current installed version. For versions v2.7.27 and earlier, you are urged to upgrade to take advantage of vastly improved filter functionality, as well as more robust syncing with O365. Versions v2.7.28 and newer should visit this knowledge base article for guidance.

IMPORTANT NOTE: *If you are experiencing lag on Discovery of an O365 Source, you may want to customize the attributes you are reading. Default settings include most of the available attributes. If you'd like to discuss Discovery performance improvement, please contact our Technical Support team for assistance.*

Finally, please note that UnitySync discovers Group objects by default to enable filtering on group membership. If you do not plan on utilizing group membership for filtering, and you want to speed up Discover, you may disable the default Discovery of Group objects.

Source Tab Configuration: Google Workspace (formerly GSuite)

As of v3.0, UnitySync can read Google Workspace Users and Contacts. The following information will help you configure your Source Tab for use to Discover objects on Google Workspace.

Google Workspace Connection Information box

Google Workspace Domain: Enter your Google Workspace domain here, then click Authorize App to obtain your Access Token and Refresh Token. You must authorize UnitySync to access your Google Workspace Domain in order to read objects.

IMPORTANT NOTE: *If you receive an error 400: invalid_request when attempting to Authorize App, please ensure you are using dirweb.exe rather than IIS to access your UnitySync UI.*

Object Types: Available object types will depend on whether you have selected *gcontacts* or *gsuite* for your Source Map Template. A selection of *gcontacts* will only allow read of contact objects, while a selection of *gsuite* will only allow read of user objects.

Optional - Filter: Filters can be used to include and/or exclude objects based on their values. Please contact Technical Support if you wish to use filters when pulling data from your G Suite Source.

Source Tab Configuration: LDIF

A typical synchronization uses one connection to read your Source LDAP directory and generate an LDIF file, then LDAP writes data from that file to sync to your Destination directory. For whatever reason, there may be instances when you may not be able to use one connection. It may be for security reasons or when the network connectivity is interrupted between your directory the UnitySync server. UnitySync can process any valid LDIF file as input. It need not have been originally generated by UnitySync.

Using UnitySync to IMPORT an LDIF File

UnitySync can use any valid LDIF file as input. This LDIF file may have been generated by another UnitySync installation, or by any other means of creating an LDIF format file.

When creating a connection to use your LDIF file as input, you will specify LDIF as the Source Sync Engine and specify the directory format that generated the LDIF file as the Destination Map Template.

By default, UnitySync will look for a file called `ldif.txt` in the connection folder for your new LDIF connection (`...\UnitySync-vX.0\Connections\MyConnection\ldif.txt`). Alternatively, the General tab includes an option to specify an alternate LDIF file location and name. You must handle getting the file into place and then removing it after it has been processed (if desired).

IMPORTANT NOTE: *Processing the same file more than once will not cause a problem. Subsequent processing of the same LDIF file will result in a sync that makes no changes to the Destination.*

There isn't a specific map template for an LDIF Source. Rather, you will need to select the map template that best corresponds with the Source that generated your LDIF, and change the sync engine to ldif rather than the default ldap.

Object Types

This is the only option that needs to be configured when reading from an LDIF Source. The available object types will differ depending on the original directory where the LDIF was created (identified by your Source Map Template).

For example, if your original Source is AD, the available Object Types will include Users, Contacts, Groups, Public Folders, and Hidden, whereas if your original Source is Notes, the available Object Types will include just Users and Groups.

When pulling from an ldif.txt file, you can identify the ldif.txt file in one of two ways:

- Copy the ldif.txt file to the root of your connections directory (i.e., ...\\unitysync-vX.0\ connections\yourconnectionname\ldif.txt)

OR

- On the General tab, specify the LDIF (Optional) location to point to the ldif.txt file you want to use as your Source.

Optional - Source Context

Just as with standard LDAP directories, you may use the Source Context field to limit the scope of items discovered from within the Source LDIF. By default, the sync will read from the root of the Source directory; in this case, all available objects in the ldif.txt file. You may choose to pull from a specific subset of this data by entering the complete DN syntax of a Source container here. For example, if your specified ldif.txt is created from the OU for Europe but you only want to sync contacts from the London Office, you'd use a format like this in the Source Context field: ou=Contacts,ou=London Office,ou=Europe,dc=domain,dc=com

Encryption of LDIF file

If your Source LDIF file is encrypted, you will need to enter the encryption password on the General tab of this connection.

If encryption of your LDIF file is necessary before moving it from Source to Destination or the reverse, you can use your own methodology. To use UnitySync's encryption feature, simply specify an encryption password on the General tab of the connection that exports the LDIF file.

Assuming you have done this on your Source connection when creating the LDIF file, you then do the same on your Destination connection and be sure to enter the same password. This will allow UnitySync to unlock and decrypt the file for processing.

Creating your own LDIF

For Active Directory/Exchange you can use the ldifde utility by Microsoft. For other directories, we suggest ldapsearch. Of course, you can also generate an ldif.txt file using an additional copy of UnitySync.

Source Tab Configuration: ODBC

Any ODBC Source (database, spreadsheet, text file) can be used as a Source of information to update an LDAP directory.

Create a new connection. On the Connection Creation pop-up, select ODBC as the Source Sync Engine type Source Map Template.

See ODBC Configuration Requirements for details on how to create an ODBC Data Source for use with your UnitySync connection.

ODBC Connection

System DSN: When using an ODBC Source you must add a DSN with the proper drivers so that UnitySync can access the data. After you have created this DSN entry, enter it here.

ID: The administrative login to the database

Password: The password that corresponds to the ID specified.

Multi Value: If any of your Source columns contain multi-valued data, each value will be separated by a special character. If you enter that separator character in the Multi Value parameter, UnitySync will parse out each value so they may be assigned to the Destination attribute individually.

IMPORTANT NOTE: *The Destination attribute mapped must be a multi-valued attribute type.*

SQL Query

SQL Query: UnitySync will execute an SQL query against the DSN. The results of this query will produce the data that will be synced. You must provide a valid SQL statement to produce the necessary data-set. Writing your own SQL statement provides you with the ability to be as broad or specific with the records you sync. For example, "Select * from employees" would provide you a data-set of all employees in your database. "Select * from employees where country = 'US' " would be a more specific and perhaps smaller data-set to sync.

Test SQL: Before you execute your sync, test your query using the Test SQL button.

Source Data

If Test SQL is successful, the Source Data will show the first few rows of output.

Field Definitions

IMPORTANT NOTE: *When a column header contains a space character, the space is converted to a dash character for the purpose of attribute definition and mapping. If the Source header is "Last Name" or ?Last_Name? you must reference this column by specifying "Last-Name", as spaces and underscores are converted to dashes in the ldif.txt file. In fact, the allowed character set on discovery of column header values are now limited to a-z A-Z 0-9 and '-' (dash).*

In a Source CSV/ODBC column header, any characters outside of this set are converted to '-' (dash). This applies to field definitions on the Source Tab, as well as mappings in your Custom Object Map or Custom List Map file.

Index: The records in the data-set being used as the Source must contain a field that holds unique information necessary to produce a unique index in the Destination. Typically this is the SMTP address or a User ID.

SMTP Address: Identify the Source attribute that contains the SMTP email address. If your custom ODBC map file uses the email address in the Destination dn, then the SMTP Address attribute must be populated for all Source records.

Group Membership: In order to create groups and apply the appropriate membership, your Source must include a column to identify the Groups each object should belong to. Specify the column that contains the Group Membership information. This should only be specified if syncing Groups as Groups. The value(s) specified in this column of your Source data will be used to 1) create the new Group(s) based on the value(s) and 2) add members to the new group.

IMPORTANT NOTE: **Since the sync is creating these groups dynamically, the new group will NOT contain an email address.*

Display Name, Personal Title, First, Middle, last, GQ: There are several ways to define a display name, and individual name parts, in the Destination directory from your Source ODBC data:

1. If you have a single displayname field in your database that contains first, last, MI, etc., then that field can be specified in the Display Name parameter. Likewise, if you you have individual fields for the individual name parts (First, Last, Middle etc) then each field name can be entered in the appropriate parameter to identify each field containing that data (First, Middle, Last etc). If your Source data does not include one of the name parts (i.e. Title, GQ) then leave it blank.

2. If you have a Display Name, but no individual name parts, you can specify the display name in the Display Name parameter, and set the Parse option to Yes under the Display Name tab. Parsing the Display Name will generate individual values for First, Last and Middle on your Destination object.
3. The logic the Parse Display Name uses is to parse the Display Name into its parts and populate the appropriate attributes. Then, format the Display Name as defined on the Display Name Options screen (First, Last, Same as Source, etc.)
4. If you do not have a single field containing a Display Name, you have two options for how to generate a Display Name:
 - If you have individual fields for all name parts, identify the name parts fields in the appropriate parameters (First, Middle, Last etc), set the Parse Option on the Display Name tab to NO and select a Display Name Format (on the Display Name tab) that uses the individual fields. i.e. Last, First.
 - Use your SQL Select statement to produce a Display Name by concatenating fields in your database (i.e. Select First, Last, MI as Displayname...).

Custom Mapping for an ODBC Source

Each ODBC Source is unique and therefore Default Mapping is of limited use. Create a Custom Object Map file as usual. The SMTP address and Name fields will be mapped successfully by default. The rest of the ODBC data Source columns must use custom mapping. See Custom Mapping: ODBC Source for specific information.

Source Tab Configuration: CSV

Using a .csv Source file is the last option we recommend, but we understand that sometimes it is the **only** option.

If you would like assistance exploring other options for your connection, please contact our Technical Support Team and we will be happy to provide recommendations for your particular needs.

That being said, it is fairly straight-forward process.

Create a new connection. On the Connection Creation pop-up, select the Source Sync Engine of CSV and Source Map Template of CSV.

Add your .csv file to the connection directory \ and rename it import.txt. Example: ...\\UnitySync-v1.x\connections\YourConnectionName\import.txt

CSV Parsing Box

Enable Column Headers: Check this option if your Source data file contains a header row. If headers are present, you will use these headers in the Field Definitions and your Custom Map File. (See Important Note under Field Definition below.) If no column headers are present, you will use default headers in the Field Definitions and Custom Map File (i.e. field0, field1, field2, etc).

Field Separator: Select the field separator used in your Source file. Supported separators are comma, tab, pipe, semicolon, colon, space.

Field Delimiter: Select the field delimiter used in your Source file. Supported delimiters are double quote, single quote, both.

Multi-Value: If any of your Source columns contain Multi-Valued data, each value will be separated by a special character. If you enter that separator character in the Multi-Value parameter, UnitySync will parse out each value so they may be assigned to the Destination attribute individually.

IMPORTANT NOTE: *The Destination attribute mapped must be a 'multi-valued attribute type.*

Source Data Box

The Source Data section displays all columns, and the first few lines of data. The data file is read and the columns displayed are based on the CSV Parsing options selected.

Field Definition Box

IMPORTANT NOTE: When a column header contains a space character, the space is converted to a dash character for the purpose of attribute definition and mapping. If the Source header is "Last Name" or ?Last_Name? you must reference this column by specifying "Last-Name", as spaces and underscores are converted to dashes in the ldif.txt file. In fact, the allowed character set on discovery of column header values are now limited to only a-z A-Z 0-9 and '-' (dash).

In a Source CSV/ODBC column header, any characters outside of this set are converted to '-' (dash). This applies to field definitions on the Source Tab, as well as mappings in your Custom Object Map or Custom List Map file.

Index field: The records in the dataset being used as the Source must contain a field that holds **unique** information necessary to produce a unique index in the Destination. Typically this is the SMTP address or a User ID.

SMTP Address field: Identify the Source attribute that contains a single SMTP email address. If your custom CSV map file uses the email address in the Destination dn, then the SMTP Address attribute must be populated for all Source records.

Group Membership field: Specify the column that contains the Group Membership information. This should only be specified if syncing Groups as Groups. The data specified in this column of your Source data will be used to 1) create the new Group(s) based on the value(s) in the column and 2) add members to the new group.

IMPORTANT NOTE: Since the sync is creating these groups dynamically, the new group will NOT contain an email address.

Display Name, Personal Title, First, Middle, Last, GQ fields

There are several ways to define a display name, individual name parts, and personal title in the Destination directory from your Source CSV data.

1. If you have a single displayname field in your database that contains first, last, MI, etc., then that field can be specified in the Display Name parameter. Likewise, if you you have individual fields for the individual name parts (First, Last, Middle etc) then each field name can be entered in the appropriate parameter to identify each field containing that data. (If your Source data does not include one of the name parts (i.e. Title, GQ) then leave it blank.)
2. If you have a DisplayName, but no individual name parts, you can specify the display name in the Display Name parameter, and set the Parse option to Yes under the Display Name tab. Parsing the DisplayName will generate individual values for First, Last and Middle. The logic the Parse Display Name approach is: Parse the Display Name into its parts and populate the appropriate attributes. Then, format the Display Name as defined on the Display Name Options screen (First, Last, Same as Source, etc.)
3. If you do not have a single field containing a DisplayName but you do have individual fields containing each name part you can enter each name part in the appropriate parameter. (If your Source data does not include one of the name parts (i.e. Title, GQ) then leave it blank.) On the Display Name format tab, set the Parse Option to NO and select select a Display Name Format that uses the individual fields.

Custom Mapping for a CSV Source

Each CSV Source file is unique and therefore Default Mapping is of limited use.

Refer to Custom Mapping: CSV Source for detailed information.

Destination Tab

The parameters available on the Destination tab will differ depending on the Destination Directory Type. Configuration options available for each directory type are outlined in separate sections, as follows.

Destination Tab Configuration: Standard LDAP Directories

LDAP directories include Active Directory, AD LDS, OpenLdap Notes, etc. The following configuration items may be available on the UnitySync Destination tab for a standard LDAP directory. If an LDAP directory does not support a particular function, the related UnitySync option will not appear on the tab. The example below shows an ActiveDir to ActiveDir-Contact (Mail-Enabled) connection.

LDAP Connection Information

IP: IP address OR hostname of the Destination system. In v4.0 and later you may specify more than one IP or hostname separated by commas to act as failover in case the first IP/hostname cannot connect.

IMPORTANT NOTE: *If you are using Join or Both, your IP or hostname MUST be a GC server for failover to work properly, and the appropriate GC ports must be opened on both.*

IMPORTANT NOTE: *All relevant connections must use the same values/order. Otherwise, you may receive a license error as UnitySync will count all the first listed IP's as separate directories. Say you are using 192.165.51.111 and 192.165.51.10. Hostnames for these servers are ServerA and ServerB. Each time you want to use list B as a failover for A, you want to be sure to list it the same way each time, and not use A as a failover for B. You can use either hostname or IP but please be consistent.*

SSL: The SSL parameter can be used to set the port, and may enable additional functionality.

- **No Encryption:** (default) sets standard port (389), no encryption
- **StartTLS:** allows encryption over standard non-encrypted port (389)
- **SSL/TLS - Basic:** sets SSL standard port (636), enables encryption but not expiration checking of certificates.
- **SSL/TLS - Expiration Check:** sets SSL port (636), enables encryption and expiration checking of certificates.

ID: The DN of an administrative account. This is required information when using Active Directory, because Microsoft directory types require an Administrative Login when creating objects. This can be left blank for anonymous access for all others. To place special permissions on this account, refer to Prepare Directory Servers: Active Directory.

PASSWORD: The password that corresponds to the Logon ID specified.

Sync Mode: Create Only

Placement DN: Exact DN of **existing** destination OU where you wish to create destination objects.

i.e. Placement DN:

```
OU=External Contacts,ou=Contacts,dc=domain.com
```

Structure Name: If no Placement DN is specified, it defaults to the root of the destination domain and a Structure Name is required. Syntax for Structure name is simply the name of the OU to sync to:

i.e. Structure Name: Outside Contacts

Without Placement DN, this will create contacts in:

```
ou=External Contacts,dc=domain,dc=com
```

**** With Placement DN**, the combined values are the destination OU where objects will be created. *Note: If the Struct Name specified does not exist on the destination, the OU (i.e. External Contacts) will be created at sync time.***

i.e. Structure Name: External Contacts

Placement DN:

```
ou=Contacts,dc=domain.com
```

The synced objects will be created in

```
OU=External Contacts,ou=Contacts,dc=domain.com
```

Sync Mode: Join and Create/Join (aka Both)

Refer to this link for details of using features Join or Both including related features Reject on Match and Modify Attribs

Sync/Join Mode: Create (Object Creation)

On the Destination tab of your connection, select a Sync/Join Mode for your connection. Your options are Create, Join or Both. The options selected will determine which of the Create/Join parameters are enabled.

Create (Object Creation)

A standard Create connection is the most commonly implemented configuration. Objects are read from the Source directory, and contacts are created on the Destination in the location configured by Structure Name and Placement DN. Future sync runs will maintain those objects created in the Sync Container - adding new objects, modifying objects already in the sync container, and processing deletes as specified.

Object Creation box

Structure Name: This parameter identifies a container name to sync objects into on the Destination. If this parameter contains a value, UnitySync will create the container in the default (root) of the Destination (unless the default is overridden by a DN in the Placement DN.) The Structure Name value should be just a name for the container; for example, *SyncContainer* or *GALContacts*.

Placement DN: By default, UnitySync will place your new container/OU in the root of the Destination directory. Placement DN is an override and will place your Destination container in the location specified here. Complete LDAP syntax is required and the tree object you specify must exist before you run the sync. Additionally, if you already have a Sync Container defined on your Destination and do not need UnitySync to create one for you, you may leave Structure Name blank and only include the Placement DN.

Syntax rules when identifying Placement DN

- **No Spaces:** Do not place spaces between "=" or ",". Spaces are allowed within the name of your element. For example, CN = My Container is incorrect. The proper syntax should read cn=My Container.
- **Case Sensitivity:** Structural objects or attribute labels (cn,dc,ou, etc.) are not case sensitive.

IMPORTANT NOTE: For AD LDS Destinations, Placement DN is **required**. For example: *ou=SyncContainer,ou=External Contacts,dc=domain,dc=com*

Deletes-Create Mode: What do you want to do on the Destination in the event that a previously synced contact is removed from the Source? The default, **Process Deletes**, means the corresponding Destination contact will be deleted. **Ignore Deletes** will do nothing to the Destination object.

IMPORTANT NOTE: UnitySync will only attempt to delete an object once. If you have Ignore Deletes set for one or more connection runs then change the option to Process Deletes, UnitySync will start processing deletes from that point forward. It will not, however, retroactively delete objects that no longer exist on the Source.

If the connection creates Users rather than Contacts, **Disable User Account** is also an option.

Sync Levels: How many levels of Source structure do you want replicated on the Destination directory? By default, 0 levels are replicated. All synced objects are placed in a single 'flat' directory as specified by Struct Name and/or Placement DN. To replicate all container levels from the Source, specify ALL instead of a number.

Trim DN: This is not a commonly used feature. Trim DN is available when Levels > 0. This option allows you to redefine the levels (container structure) that are replicated to the Destination. The actual components available for trimming as well as the actual Trim results will vary depending on your environment and connection configuration.

The best way to determine if Trim DN is useful in your connection is to run Discovery and Simulation, then review the log file to review the container structure of each object as it is added. Depending on the type of object being created, there may be additional configurations required. More information on Trim DN is available in our knowledge base.

Sync/Join Mode: Join (Join with Existing Objects)

If your Destination directory already contains person objects, you can configure a connection to search for and link to just these existing objects. This is handy when you have existing objects that need to be updated with a new attribute, among other reasons.

A join query is required for each object type to identify matches between the Source and Destination. In a Join Only connection, objects that already exist on the Destination are updated, but no new objects are created and objects are never deleted.

Join with Existing Objects box

Joins are based on a join query. This allows LDAP query logic to be specified in order to compare multiple values to identify objects to match. You'll need to enter a valid LDAP query. to compare one or more Source attribute values to one or more Destination attribute values.

Example: Query on mail and proxyaddresses

We recommend comparing the Source object's mail and proxyaddresses to the Destination object's mail and proxyaddresses, allowing a join to match on existing objects.

Recommended join query syntax: `(!(mail=^mail^)(proxyaddresses=[proxyaddresses]))`

Translation: (Dest mail = Source mail) -OR- (any Dest Proxyaddresses = any Source Proxyaddresses)

If you are not able to use the recommended query above for whatever reason, please contact our Technical Support Team for assistance in building a custom join query.

Make sure to populate each of the object type query fields that you are syncing from the Source. If you have Users, Objects and Groups checked on the Source tab in the Object Types section, you should have a query populated for User(s) Query, Contact(s) Query and Group(s) Query under Join with existing objects on the Destination tab.

Base DN (Optional): By default, the Join will begin searching the root of the Destination LDAP directory. You may enter a more specific location to narrow the search. Syntax rules for Base DN are the same as Placement DN in the Object Creation box. Complete LDAP syntax is required.

Syntax rules when identifying Base DN

- **No Spaces:** Do not place spaces between "=" or "," Spaces are allowed within the name of your element. For example, CN = My Container is incorrect. The proper syntax should read cn=My Container.
- **Case Sensitivity:** Structural objects or attribute labels (cn,dc,ou, etc.) are not case sensitive.

Sync/Join Mode: Both (Create Objects AND Join with Existing Objects)

In this configuration, the Create and Join functions are combined. See the other information which describes the individual Create and Join configuration and options. You'll find each box - Object Creation and Join with Existing Objects - and many of the same options. We'll explain those options that are specific to this mode below.

When a Both (Create/Join) connection runs, it will search for an existing Destination object based on your Join criteria. If an existing object is found, it may be updated depending on the options you have selected. If no existing object is found, a new object will be created based on the Create criteria. Unlike Join only connections, if a Source object is deleted (or falls outside the scope of the sync) the corresponding Destination object may be deleted depending on the options you have selected.

Deletes - Both (Create/Join) Mode

How do you want to handle deletes inside and outside the sync container?

You'll notice that when you select a Sync/Join mode of Both, Delete options are offered in the Join with Existing Objects section as well as in the Object Creation section.

The Delete options in the Object Creation section refer to objects that fall inside the Sync Container as specified in the Structure Name or Placement DN field, and works just the same as described for Create only Sync/Join mode. See the Deletes section in Create Mode to refresh your memory.

Delete options offered in the Join with existing objects section, however, refer to objects that exist **outside** the sync container specified in the Object Creation section. By default, this is set to *Ignore Deletes* which means that objects deleted on the Source which do NOT reside in the Sync Container will not be affected.

Process Deletes means that objects deleted on the Source which do NOT reside in the Sync Container **WILL** be deleted. This option is not often used. Most clients want objects that already existed on the Destination prior to initial sync to be unaffected if the corresponding Source object is deleted or moved outside the scope of the Sync.

Finally, you may elect to *Disable User Account* of an object that exists outside the Sync Container if it is deleted on the Source. Again, this option is not often used.

Important Notes about Joining With Existing Objects on Sync/Join Mode Join and Both

- When Joining with objects on an AD/Ex200x domain, UnitySync will query the Global Catalog to find the match. Therefore, the connection's Destination IP must be that of a Domain Controller that contains the Global Catalog. Always specify the standard LDAP port (389) when writing to AD. However, since the Join will query the Global Catalog, the UnitySync server must have access to read from the Global Catalog's LDAP port, 3268.
- The attribute specified as in the Join query must be in the Global Catalog and must be indexed.
- Join mode will find and link to any type of Person object (i.e. User or Contact). As long as the index matches, a link is established. If the matching index is a User object, that User object will be linked. If the matching index is a Contact, that Contact will be linked.

*If Create function is being used in conjunction with Join, all Joined objects become 'owned' by UnitySync. If Delete Processing is set to 'Process Deletes' these objects may be deleted by UnitySync. If you want straight Join functionality (that will never delete your existing objects) you should not specify any Create parameters. Alternatively, you can set Process Deletes to 'Ignore Deletes'.

If you have any questions about Sync/Join Mode and what might be the best options for your environment and goals, please contact our Technical Support Team.

Destination Tab Configuration: Standard LDAP Directories (continued)

Attributes Box

Modify Attributes: By default, all modifications of synced objects will use the default or custom map files to define which attributes will be monitored for changes on the Source and updated on the Destination. You may elect to look for changes on only specific Source attributes and update only those attributes on the Destination. To do this, use a comma delimited list in the Modify Attributes parameter. If this parameter contains one or more attributes, all Modifies applied to Destination objects will be limited to the attributes listed here.

IMPORTANT NOTE: *This does not apply to Creation of new objects. Creation of new objects will always apply the full compliment of mapped attributes.*

List Processing

Syncing Groups as Email Addresses

The default option for syncing your Source Groups/Lists is *Process Lists as: As Email Addresses*. This means that only the email address of the list is synced, not group membership information.

Syncing Groups as Groups

The ability to sync Groups as Groups will depend on your Destination directory type. You'll only see the option in the List Processing drop-down list when it is available.

When you sync Groups as Groups, a Group object will be created on the Destination and membership information will be synced. This is also referred to as List Processing.

For Example: When syncing to an Active Directory Destination, you may sync to create any one of the following types of AD Groups:

- Global Distribution Group
- Global Security Group
- Universal Distribution Group
- Global Distribution Group (Mail Enabled)
- Global Security Group (Mail Enabled)
- Universal Distribution Group ((Mail Enabled)

IMPORTANT NOTE: *Mail Enabled Groups are only valid if your Destination Active Directory has the Exchange schema loaded.*

Notes Regarding Syncing Groups as Groups

- Generally, the default syncing 'Groups as Email Addresses' meets the needs of most Group sync requirements. Before implementing Group as Group sync, please evaluate if it is really needed.
- By default a connection that syncs Groups as Groups will only maintain members whose User/Contact record is also synced by that same connection. If a Group member is not synced, that member will be left out of the synced Group. This is because each connection can only resolve Membership DN's of the objects it syncs. This may leave your Membership list a little short if some of the member contacts were synced in via UnitySync and then added to Groups on the Destination server. If those Groups are synced back to the originating server, they will be missing those members because the contacts are excluded from the reverse sync. There is an option that may be used to allow these Memberships to be maintained. Contact support@dirwiz.com for more information if you have this need.
- Manually created Source User/Contacts (aka not created by any UnitySync connection) that reside in containers that are EXCLUDED from the sync will always be dropped when syncing Groups as Groups. This is because UnitySync has no way of knowing if there is a valid contact for that member on the Destination or what it's DN might be.
- When a Group is initially synced, the member list may be incomplete. A subsequent sync will get the new Group object's membership list up-to-date. This occurs because the Groups member list can only contain objects that are included in this connection's sync. Sometimes, during the sync process, the Group object will sync over before it's member objects. When this happens, those members who have not yet synced will not be included in the Group. The next sync will bring the membership up-to-date.
- The display name's of Groups are synced over exactly as they appear on the Source. Reformat Display Name option on the Display Name tab does not apply to Groups. You have the option of specifying a label to be appended to the Display Name of your Group object. This label may be specified in the Display Name tab, in the Display Name Label section's Group parameter.
- If your Source is other than LDAP (i.e. CSV, Fixed, ODBC, Oracle, etc) you only have the option of syncing Groups as Groups. Your Source data must contain a column identifying the Group(s) you want each record to belong to on the Destination.

IMPORTANT NOTE: *In this case, since we dynamically create the Destination Group object, it will NOT contain an email address.*

Destination Tab Configuration: Office 365 (O365)

Options for Office 365 (O365) are a simplified version of standard LDAP connections. Please note that connections to an O365 directory can only be run in Create mode, and Join functionality is not available.

O365 Connection Information Box

ID: The login of the default administrative account. This is required information when using O365. **Password:** The password that corresponds to the Login ID specified.

Object Creation Box

What do you want to do on the Destination in the event that a previously synced contact is removed from the Source?

The default, **Process Deletes** means the corresponding Destination contact will be deleted.

Ignore Deletes will do nothing to the Destination object.

IMPORTANT NOTE: *UnitySync will only attempt to delete an object once. If you have Ignore Deletes set for one or more connections then change the option to Process Deletes, UnitySync will start processing deletes from that point forward. It will not, however, retroactively delete objects that no longer exist on the Source.*

Sync Lag on O365 Destination

Please be aware that objects synced to O365 - the cloud - may not be immediately viewable. This applies to object creation, modification and deletion. We observed this behavior during testing with just a few objects and are continuing to test with more objects to see if there is any correlation between the number of objects synced and time before updates are able to be viewed. If it has been more than a few minutes and you still do not see new or updated objects on the O365 Destination, please contact our Technical Support Team for troubleshooting assistance.

Also, please be mindful of the lag when performing subsequent syncs or un syncing. Allow at least a few minutes between connection runs before attempting to sync or un sync.

Destination Tab Configuration: Google Workspace (formerly G Suite)

As of v3.0, UnitySync can write Contacts to Google Workspace. The following information will help you configure your Destination Tab for use to Synchronize objects to Google Workspace.

Google Workspace Connection Information box

Google Workspace Domain: Enter your Google Workspace domain here, then click Authorize App to obtain your Access Token and Refresh Token. You must authorize UnitySync to access your Google Workspace Domain in order to write objects.

IMPORTANT NOTE: *If you receive an error 400: invalid_request when attempting to Authorize App, please ensure you are using dirweb.exe rather than IIS to access your UnitySync UI.*

Object Creation box

Deletes: You have the choice to Process or Ignore Deletes. Process Deletes means that UnitySync will delete objects on the Destination if they fall out of the scope of the Source Discovery, either because they've been deleted, met a filter condition, or moved from the specific Source Context. Ignore Deletes will leave those objects that fall out of the scope of the Source Discovery active on the Google Workspace Destination.

IMPORTANT NOTE: **Contact objects on a Google Workspace Destination exist in a flat file - there is no structure. Before Syncing to a Google Workspace Destination, you may want to configure a Discovery first, to ascertain if any contact objects already exist. It is also important to note that while UnitySync can detect orphans on a Google Workspace Destination, it will read ALL orphans, and not only those objects originally put there by UnitySync. So, knowing what objects may exist before populating a Google Workspace Destination may be helpful in the long term.*

Destination Tab Configuration: ODBC

ODBC Connection Box

System DSN - Select a DSN (Data Source Name) from the available list of DSNs. If you do not yet have a DSN for your odbc source, refer to the Configuring ODBC topic for information on how to create a System DSN.

ID - If your DSN references a password protected database, enter the necessary Login ID.

Password - Password for Login ID specified above.

Destination Table Information

Table Name - Specify the name of the database table to write to.

Index Field - Specify the name of the table's index field.

IMPORTANT NOTE: *If objects already exist in your Destination database, this index field is used to identify matching entries between Source and Destination. The Destination attribute to compare against is the one mapped to your Index field in your Custom Map file (i.e., YourIndexField=^SourceAttribute^).*

Important Notes about your Destination Database

The Destination database must exist and the table you write to must also exist and have all columns defined. UnitySync does not create the database, table or columns.

Custom Mapping is required. See our knowledge base article on how to create a custom object map file for detailed instructions. Also see Custom Mapping: ODBC Destination topic for details specific to an ODBC Destination.

It is highly recommended that this index field be configured in the Destination ODBC database as "Duplicates Allowed = No" otherwise, duplicate Source data may result in duplicate database entries. Also, the UnitySync Recovery Procedure will not process successfully unless "Duplicates Allowed = No".

When syncing to a database via an ODBC destination, you may or may not need to specify the Destination index as a NO-Mod-Attrib (depends on your databasetype and column settings). *So, begin your testing without this setting.* If you receive database error on Mod attempts during testing, return here to add this setting:

To do this, edit the Raw Config from the Custom tab. Simply click the Raw Config button on the Custom tab, and add the following line:

```
no-mod-attrs=IndexAttributeName
```

Add Function Box

Select how you want to handle new data from the Source. If you want to update the existing ODBC database with new objects, select Process Adds. If you only want to modify existing ODBC objects, select Ignore Adds and no new objects will be added, only existing objects will be modified with any changes on the Source.

Delete Function Box

Select what you want to do when an object is deleted from the Source. If you'd like the database to remain the same despite deletions on the Source, select Ignore Deletes. If you would like objects deleted on the source to also be deleted in the ODBC database, select Process Deletes.

Destination Tab Configuration: LDIF

You may use UnitySync to generate an LDIF file for use by another UnitySync connection or for any other use that requires and LDIF formatted file. When creating your LDIF Export connection, pick the appropriate Destination Map Template that represents the format you wish to write in the LDIF file. Then, you will specify *ldif* as the Destination Sync Engine.

UnitySync Discovery will read your LDAP Source and write to a 'raw' LDIF file called *ldif.txt*. This LDIF file will include all attributes read from the Source directory. If the raw output is what you want, that's all you need to do. The LDIF file is ready for use as an LDIF input Source. There is no need to run the Sync phase of this connection.

If you wish to Sync your 'raw' LDIF data through a map file to produce another LDIF file that contains only specific attributes, you would then run the Sync phase. The UnitySync Sync phase will generate an LDIF file named export.txt. The Sync process will use default or Custom Mapping to export a specific set of attributes for each record exported to the LDIF export.txt file.

Whether you use the raw LDIF file produced by Discovery, or the mapped export.txt produced by the Sync, you can use FTP, Email, or some other means of transport to place the file on a machine where it can be used as input.

Below are optional parameters that may be specified on the Destination tab of a connection that exports to an LDIF file. These parameters may be combined to format the DN of each object exported to the LDIF file. You may specify a value for one, two or all parameters.

Object Creation Box

Structure Name: This is a simple container name that will be used as the structure in the DN of each exported record. For example, if the value of SyncContainer is entered, then each object will export with a default dn as follows:

```
dn:cn=joe@abc.com,ou=Structname,dc=domain,dc=com
```

Sync Levels: How many levels do you want represented in the Destination DN? Following on the above example, if you leave Sync Level blank or set it to 0, you will get the same as the above example. if you set Sync Level to 1, each export object will export with a dn as follows:

```
dn:cn=joe@abc.com,ou=1stLevel,ou=Structname,dc=domain,dc=com
```

If you set Sync Level to ALL, you will receive all levels of structure from the Source.

Trim DN: This is not a commonly used feature. Trim DN is available when Levels > 0. This option allows you to redefine the levels (container structure) that are replicated to the Destination. The actual components available for trimming as well as the actual Trim results will vary depending on your environment and connection configuration. The best way determine if Trim DN is useful in your connection is to run Discovery and Simulation, then review the log file to review the container structure of each object as it is added. Depending on the type of object being created, there may be additional configurations required. More information on Trim DN is available in our knowledge base.

Placement DN

Exact DN structure to build. Any value entered in the Structure Name will be combined with this Placement DN. Continuing on the above example, if you also add a Placement DN of ou=Recipients,dc=domain,dc=com, each object will export with a dn as follows:

```
dn:cn=joe@abc.com,ou=Structname,ou=Recipients,dc=domain,dc=com
```

List Processing

List Processing options are available for syncing to an LDIF Destination, but not recommended. If the default method of syncing Groups as Email Addresses does not suit your needs when syncing to an LDIF, please contact Technical Support for guidance.

Encryption of LDIF File

If encryption of your LDIF file is necessary, review information in the LDIF Source section of this guide.

Creating Your Own LDIF

Instructions are found in the LDIF Source section of this guide.

Destination Tab Configuration: CSV (Text)

Custom Map Files for connections outputting to a text file are different than those used for LDAP to LDAP connections. An LDAP to LDAP connection map file is used to define Source to Destination mapping. For text output connections,

the map file is used to define the format and content of the text output file. You can choose from an available list of separators and delimiters.

Note that there is not a specific map template for CSV output. Select the most appropriate map template to obtain the attributes you desire, and change the sync engine to csv.

Your sync results will reside in a file called export.txt found in the connection folder:

```
...\UnitySync-v1.x\connections\YourConnectionName
```

Special Characters Box

Field Separator: Available separator characters are tab, comma and the pipe symbol. Select comma to generate a comma separated file (csv).

Field Delimiter: Available delimiters are double quote and single quote.

The combination of separator and delimiter you chose depends on your Source data and what you plan to use the text output file for. If you are using it as input for another process, be sure to configure the text output to meet any requirements of that process.

Column Headers Box

Enable Column Headers: You may elect to enable or disable column headers in your output text file. If you want a row of column headers as the first line in your output text file, then Enable Column Headers.

Custom Mapping for CSV Output

It is likely that you will always want to utilize Custom Mapping when using a connection that syncs to a CSV Output file. Please see Custom Mapping: CSV Destination for more information.

Destination Tab Configuration: DNHASHGEN

This is a very unique type of connection. The Destination Sync Engine type of DNHASH is only used in a few special cases.

This type of connection will perform a *Do Nothing Join*. The *Do Nothing Join* reads the Source, uses the indexes provided to identify the matching record on the Destination. No changes are applied to the Destination objects. Instead, all that happens is a file named export.txt is generated as output.

For more information on this topic, see the topic in our knowledge base or contact our Technical Support Team.

Display Name Tab

This tab contains options for manipulating the Display Name format.

Display Name Label Box

Object field: You may opt to append a label to the Display Name of all synced objects. The label specified here will be appended to each display name. You may specify a label for one, two or all object types. They may be the same or different depending on our needs. These labels can be used as a method to distinguish names by their original Source. No leading space is necessary, one will be added automatically. This is simply a label and not a full DN.

Example: [DirWiz Contact] *Result:* Smith, John Q [DirWiz Contact]

Group and Folder fields: Same as above, but only applied to Groups or Folders, respectively.

Parse Display Name Box

By default, 'Parse Display Name' is disabled (No). When disabled, the actual Source name part attributes (sn, givenname, initials) are used to set the Destination name part attributes. When you enable Parse Display Name, UnitySync parses the incoming Source Display Name and will attempt to separate the Display Name into its component parts (sn, givenname, initials). It is these name parts, pulled from the Source Display Name, that are then used to set your Destination name part attributes (sn, givenname, initials). The Destination Display Name will be formatted based on the format you have selected in Reformat Display Name. If you have Parse Display Name enabled, the Display Name will be formatted using these name parts pulled from the Source Display Name.

Adaptive

This option is disabled by default. This is not a commonly used option. This option aids in parsing Source Display Names whose format is outside the scope of what the standard Parse function can comprehend. Adaptive is only available when Parse Display Name is enabled. When Parse and Adaptive are both enabled, the sync will Parse the Source Display Name as usual. Then the Adaptive feature will compare the name parts obtained by the Parse function against the incoming Source attributes (sn, givenname, initials). This allows us to be sure we have properly identified the name part values, thereby helping in the parsing of the remaining values in the display name. This option will commonly be used with the options available in Advanced Parsing (see below).

Reformat Display Name box

Display Name Format

This configuration option allows you to manipulate the display name field. It will determine how names appear in your Destination directory. Each option produces a different format.

IMPORTANT NOTE: *This configuration item does not apply to Groups/Lists. Groups/Lists are always synced using the same display name as appears on the Source.*

Options available in the UnitySync UI:

Option	Result
Same as Source	Leave as appears on Source
<first> <last>	John Smith
<last>, <first>	Smith, John
<last> <first>	Smith John
<last>,<first>	Smith,John
<First> <MI> <Last>	John Q Smith
<Last>, <First> <MI>	Smith, John Q
Blank field (Custom)	Enter your desired format

The predefined Reformat Display Name options allow you to select from a list of commonly used formats. However, if

you require something other than these options, you can use the Custom option to build the Destination displayname in a way that will meet your requirements.

Custom

The custom Displayname formatting option allows you to define the exact attributes/format you want to use to build your Destination displayname. You may use `^Source^` attributes or you may use `~internal~` attributes. You may also include hard coded spaces, commas and/or text. The syntax is similar to that used on custom mapping. Below are some examples.

Example #1: DoD AD Source standard, with Parse enabled

```
~sn~, ~givenname~ ~initials~ ~gq~ ~personaltitle~ ^extensionattribute4^ ^company^ ^department^
```

Translates to: Last, first MI GQ rank/title Nationality DoDcomponent DoDsubcomponent

Example #2: DoD Ex55 Source standard, with Parse enabled

```
~sn~, ~givenname~ ~initials~ ~gq~ ~personaltitle~ ^extension-attribute-4^ ^company^ ^department^
```

Translates to: Last, first MI GQ rank/title Nationality DoDcomponent DoDsubcomponent

Example #3: Standard with title, company, and department

```
^sn^, ^personaltitle^ ^givenname^ ^initials^ - ^title^ - ^company^, ^department^
```

Translates to: Smith, Mr. James, J - Sales Manager - Directory Wizards, Sales

Example #4: Append label before displayname

```
[External] ^givenname^ ^sn^
```

Translates to: [External] Sally Jones

Global (optional configuration)

The Global Displayname option is pre-configured to the format shown below. This is the DISA/DoD default displayname format, for pulling from an AD Source directory.

This global setting is not available in the UnitySync UI, but is configurable via the `...\global\displayname.txt` file. Similar to the Custom option outlined above, the Global option allows you to configure this default format by modifying the `...\global\displayname.txt` file.

In the current version, this override file will be used for a connection if you manually change the Raw Config (on the Custom tab) parameter to:

```
display-firstlast=global
```

Default Global format:

```
~sn~, ~givenname~ ~initials~ ~gq~ ~personaltitle~ ^extensionattribute4^ ^company^ ^department^
```

IMPORTANT NOTE: This global setting modifies ALL connections in this installation of UnitySync. If you want to adopt this displayname format for only SOME connections in your installation, use the Global format as defined above in the Custom field for each desired connection.

Display Name Tab: Advanced Parsing of Custom Source Display Names

In some cases, the incoming Source Display Name is customized as to include data other than First, Last, MI. This additional data may include custom titles, organizational groups, or special company specific codes to identify subsets of users. When using Parse Display Name and Custom Mapping, you can configure special custom files that will allow you to extract this display name data into internal variables. These files exist in the `\global` directory:

- custom.txt
- junk.txt
- title.txt
- GQ.txt

You will find default versions of these files already exist. These files are applied globally, to all connections when Parse Display Name is enabled.

Advanced Parsing Details

When Parse Display Name is set to Yes, the incoming displayname is parsed to obtain individual name parts including First, Last, Middle Initial, Personal Title, and Generational Qualifier. These are set to internal variables `~givenname~`, `~sn~`, `~initials~`, `~personaltitle~` and `~gq~`.

Sometimes, your incoming Source displaynames have been customized to meet company standards and may include other components you wish to parse out at sync time. There are a few configuration files that may be modified to help achieve the appropriate displayname parsing for your unique environment.

IMPORTANT NOTE: *These options apply to any displayname setting other than 'Same As Source'. If 'Same As Source' is selected, none of these Advanced Parsing configurations will take effect.*

Title.txt

The default title.txt file contains a list of the most common titles (i.e. Mrs, Mr, Ms) as well as many of the most common military titles. You may add additional titles as necessary. At sync time, any value found in the incoming displayname that matches a defined Title will be saved in the variable `~personaltitle~`. Multiple titles will be appended if necessary.

Custom.txt

The default file is basically empty. You may add any additional values as necessary. This file may be used to capture custom displayname notations. At sync time, any value found in the incoming displayname that matches a defined Custom value will be saved in the variable `~custom~`. Multiple values will be appended if necessary.

Junk.txt

Sometimes, the presence of custom notations not included in Title.txt or Custom.txt can confuse Display Name Parsing. If your Source Display Name includes custom values that are not added to Title.txt or Custom.txt they may need to be added to Junk.txt. When using Display Name Parsing, values in Junk.txt are extracted and thrown away, not saved to a variable and not included in the Destination Display Name format (when using anything other than Same As Source).

GQ.txt

The default GQ.txt file contains a list of the most common Generational Qualifiers (i.e. Jr Sr III Esq etc). You may add additional GQ values as necessary. At sync time, any value found in the incoming displayname that matches a defined GQ will be saved in the variable `~GQ~`. Multiple GQs will be appended if necessary.

IMPORTANT NOTE: *Any value in the Source Display Name that isn't part of recognized Name Parts, or included in the Advanced Parsing files (Title.txt, Custom.txt, Junk.txt) will be appended into an internal variable `~leftover~`.*

SMTP Addresses Tab

When syncing to create mail enabled objects on the Destination, the default mapping and configuration will sync over the primary SMTP and secondary smtp addresses exactly as they appear on the Source.

There are three configuration options available that allow you to manipulate the Destination proxy addresses. These are Drop Proxy, Substitute Domain and Swap Proxy. Each option is outlined below, including usage examples.

If more than one option is configured, the options are applied in the order they appear in the GUI, that is:

- **Drop Proxy:** Any domains specified by Drop Proxy are dropped, first
- **Substitute Domain:** Substitution of the primary SMTP domain is applied.
- **Swap Proxy:** Any necessary swapping is performed.

SMTP Addresses Tab: Drop Proxy

When you run UnitySync to Discover objects that contain secondary proxyaddresses, and Sync to create mail enabled objects that contain secondary proxyaddresses, you have the option to drop one or more incoming proxyaddress(es). Just provide one or more domain names in the Drop Proxy field. The format is @DomainToDrop.com (ex. @alpha.com,@beta.org), comma separated without spaces after the comma. When the Destination contact is created, its proxyAddresses will not include any address with a Drop Proxy domain.

Examples of Drop Proxy Usage

If the incoming secondary proxyaddresses of a Source object are:

```
joe@alpha.com
joe@beta.org
joe@abc.com
joe@any.local
joe@anyother.local
joe@child.beta.org
joe@xyz.com
```

And you've listed the following domains in your Drop Proxy field:

```
@alpha.com,beta.org,local
```

The new Destination contact will contain only the following proxyaddresses:

```
joe@abc.com
joe@xyz.com
```

A Note about Drop Proxy syntax: It may or may not be necessary to include the @ prefix in the Drop Proxy domains you specify. An example of when it is necessary is if you have two incoming domains that look like this:

```
joe@child.beta.org
joe@beta.org
```

If you specify a Drop Proxy of simply beta.org, then both of the above addresses will be dropped. If you wish to ONLY drop the Joe@beta.org address, then your Drop Proxy value should include the @ as follows:

```
@beta.org
```

However, if you want to drop any proxyaddresses that contain beta.org, including child.beta.org, your Drop Proxy value should not include the @ symbol:

```
beta.org
```

SMTP Addresses Tab: Substitute SMTP Domain

There may be an instance when you want to replace or substitute the incoming primary SMTP proxyaddresses domain name when creating a new Destination object.

Enter the desired domain name in the Substitute Domain parameter. All incoming primary addresses will be modified to substitute the specified domain name.

Example of Substitute Domain Usage

Substitute Domain will replace the domain name of the incoming primary SMTP address.

Your Substitute Domain value is:

NewDomain.com

Three incoming objects have primary SMTP addresses as follows:

mmouse@disney.com
 jcarter@whitehouse.gov
 slee@marvel.com

When the parameter is set with a new domain (ie. dirwiz.com) the following addresses will be created:

mmouse@NewDomain.com
 jcarter@NewDomain.com
 slee@NewDomain.com

Notes about Substitute Domain Syntax

- Only a single Substitute Domain may be specified.
- It is **NEVER** appropriate to include the @ prefix when specifying your Substitute Domain. If you include the @ prefix, you will get address values that look like this:

bbunny@@NewDomain.com

SMTP Addresses Tab: Swap Proxy

When syncing LDAP directories, you have the option to manipulate proxyaddress, swapping a secondary **smtp** up to be the primary ***SMTP****.

Typically an Exchange mailbox is given a primary **SMTP** address. This address is the user's reply-to address. Secondary **smtp** addresses can be added that allow the user to receive email using other addresses.

UnitySync's Swap Proxy function reads through the proxyaddresses using a rule-set. If a match occurs, that address becomes the primary address and the old primary address is pushed into the proxy list.

The swapping logic looks like this:

- Read the domain part of each proxy addresses and compare, in order, against those you have supplied in the Swap Proxy parameter.
- When a match is found:
 - Swap the matching **secondary smtp** proxyaddress, making it the Primary **SMTP** address.
 - AND**
 - Swap the original Primary **SMTP** address making it a secondary **smtp** proxyaddresses

Note, when a Swap takes place, **mail** and **targetaddress** are also set using the swapped address.

Example of Standard Swap Proxy Usage

Assume the following three records exist on your Source.

SMTP:jsmith@aol.com
 smtp:james@zoomit.com
 smtp:support@p.com
 smtp:jim@home.com

SMTP:tsmith@aol.com
 smtp:tom@zoomit.com
 smtp:tommy@home.com

SMTP:bubba@slow.net
 smtp:Charles@fast.net

On the ProxyAddresses tab, the Swap Proxy field is set to:

- @p.com,@home.com

When a match is found:

- Swap the matching **smtp**, making it the primary **SMTP** address.

Resulting three Destination Records:

SMTP: support@p.com
 smtp:james@zoomit.com
 smtp:jsmith@aol.com
 smtp:jim@home.com

SMTP:tommy@home.com
 smtp:tom@zoomit.com
 smtp:tsmith@aol.com

SMTP:bubba@slow.net
 smtp:Charles@fast.net (nothing is swapped)

Notes about Swap Proxy Syntax

It may or may not be necessary to include the @ prefix in the Swap Proxy domains you specify. An example of when it IS necessary is if you have two incoming domains that look like this:

EricN@DomainXYZ.com
 Eric@XYZ.com

If you specify a **SwapProxy** of simply **XYZ.com**, then both smtp addresses above qualify for swapping. Which ever is returned first will be swapped to the primary. If you specifically want the @XYZ.com domain to be primary, then you need to specify the Swap Proxy value as **@XYZ.com**.

There are some guidelines to the rule-set:

- The order of the swap proxy list sets the order of the search through the secondary addresses.
- The swap proxy list is done by substring matching (no wildcards should be used)
- If a proxy is not matched the addresses are left alone.
- UnitySync will use the first proxy it matches.

SMTP Addresses Tab: Swap Proxy - SwapTargetOnly

The SwapProxy feature allows you to swap mail, targetaddress and proxyaddresses in favor of a specific email domain. The solution requires the originating source object to include a secondary smtp address of the desired domain. The secondary and the existing primary will be swapped (hence the name SWAP).

There may be a circumstance, however, where you want to swap just the targetaddress at Sync time.

To do so, implement SwapProxy as usual, then customize your map file to ensure your DN, mail and proxyaddresses do not get swapped.

1. Create a custom object map file
2. Change the **DN** mapping to use `^mail^` instead of `~mail~`
 i.e. `dn=cn=^mail#64^,~struct~`
3. Change the **mail** mapping to use `^mail^` instead of `~mail~`
 i.e. `mail#256=^mail^`

4. Change the **proxyaddresses** mapping to use [allmail] and add a new proxyaddresses line above it to define SMTP. Exact syntax is:

```
proxyaddresses=SMTP:^mail^  
proxyaddresses=smtplib:[allmail]
```

5. The default **targetaddress** mapping should remain **unchanged**. It uses ~target~ which is variable based on swap proxy.

```
targetaddress=~target~
```

6. If you need additional proxyaddresses like SIP or x500, see this knowledge base article for the specific mappings.
7. Save the file.

Run a new Simulation. You should see that with the above modifications, only targetaddress gets modified.

Custom Tab

Map Files box

UnitySync mapping files dictate the movement of objects and attributes between the Source and Destination directories. Default attribute mapping is pre-defined for each connection through the use of default map files located in the ...\\UnitySync-vX.0\\global\\map directory. **Do not edit these default map files! These files are distributed with UnitySync and will be overwritten with future upgrades.** These default map files are defined for specific object/attribute types. Most commonly used attributes are mapped from the Source to the Destination.

Default map files include:

Object Map

This file defines the attribute mapping for each Person/User object.

List Map

Applicable to connections syncing Groups and Group Membership, rather than just syncing the email address of Groups. This file defines the attribute mapping for each Group/List object.

Folder Mappings

This file defines the attribute mapping for each Folder object.

Sourcedef

This file defines the Source structure and person objects as well as directory specific configurations required by UnitySync.

Struct Map

This defines the Custom Tab attribute mappings for each structural element automatically generated by UnitySync (containers, OU's). This is a very rarely used feature.

Struct Map 1

Applicable to connections with a Destination type of Notes only. This is used in special cases where a one time root object needs to be written. In the Notes case this defines an organization root object where all other structure objects are organizational units. This is a very rarely used feature.

If you have need to alter the default mapping, visit the Custom Mapping section of our Administrator's Guide or the Custom Mapping topic of our knowledge base for more information.

Miscellaneous box

The Miscellaneous box options provide you the means to edit other configuration items.

Raw Config

This allows you to edit your config.txt file from the UnitySync UI. The config.txt file is the heart of UnitySync. It contains all of the details of your configuration as set up in the UnitySync Manager UI.

You should only make adjustments to the Raw Config when directed to do so by Directory Wizards Technical Support. When directed, you can edit the Raw Config using this button.

Lookup

This functionality allows you to provide a table of values for string replacement on attribute mapping.

Examples of uses for Lookup include:

Custom Macro Mapping using a Lookup Table

Country to Country Code Lookup Table

See Custom Map Files for more information.

Excludes

This list defines exclusions of Source object based on DN or attribute value. Exclusions are very important to the directory synchronization process as they are used to filter or limit the information you move between directories. See Exclusions for more information on using Excludes.

Internal Variables

During the Sync process, UnitySync generates Internal Variables for use in default mapping files. The Internal Variables option on the Custom tab allows you to define your own Internal Variables unique to each connection, then use those defined variables for custom mapping purposes.

Download

export.txt

This button allows you to download the export.txt file created when you sync to CSV.

ldif.txt

This button allows you to download the ldif.txt file created on Discovery.

Email Notify Tab

One way to stay abreast of your directory sync status is to configure Email Notifications to send status reports after each sync is performed. Using this option you can email an administrator a notification on every run of the software (regardless of connection) or if there are problems with a sync. In addition the log files can be sent as attachments.

You already know about global notifications via the Global SMTP button in the TopRow. You can also set up different notifications per connection. If you use the Email Notify tab within your connection, both global and specific connection notifications will be sent.

Email Notification Settings

To enable UnitySync Email Notifications either globally or per connection, Select Global SMTP from the TopRow. You must specify an SMTP Server and Reply Address to enable Email Notifications in general and in order to utilize connection specific emails via the Email Notify tab.

If you enter one or more email addresses in the Email Notification's Summary and Error parameters, then an email will be sent to these recipients upon the completion of every sync by default. Alternatively, you may elect to set recipients on a per connection basis (see below). Or, you may use a combination of both global and per connection notifications. If you do not elect to set global recipients, you must still specify the global SMTP Server and Reply Address in this Global SMTP area in order to set recipients on a per connection basis.

You also have the option for Summary Email Notifications to check Add/Mod/Del Only. Normally, if Summary notification is enabled, an email notification for every Discovery/Sim/Sync is sent. Selecting Add/Mod/Del Only enhances the Summary notification option; an email notification is never sent for Discovery and will only be sent on Simulation/Synchronization if an Add, Mod, or Delete occurs on the run.

In essence, when this this option is selected, no news is good news as you will not receive a notification if no changes are logged during the Simulation or Synchronization.

SMTP Server Settings - Global SMTP

These settings are required to enable email notifications.

SMTP Server

TCP/IP address (or host name) of a SMTP server. This is required information. If left blank, all SMTP notifications will be disabled.

Reply Address

This is a standard SMTP address such as: support@dirwiz.com. This address is included in the message sent from UnitySync.

Email Notification - Global SMTP and Email Notify tab

These values are optional. The recipients may be defined globally via Global SMTP (described here), or they may be set within each individual connection on the Email Notify tab.

Summary

On every run of UnitySync a summary report of the sync results will be sent to this address as a low priority message. Multiple addresses can be separated by a comma. If this field is left blank or the configuration parameter is missing, no summary E-Mails will be sent by default.

Recipients entered on a per connection basis will still receive notifications.

Add/Mod/Del Only

Normally, when you select Send Log (below), a summary email is sent for each action, whether or not anything happens during the Simulation or Synchronization runs. When you select this option instead, an email notification is never sent for Discovery and will only be sent on Sim/Sync if an Add/Mod/Delete occurs on the run.

Error Email

If UnitySync encounters a synchronization error a summary report of the sync results will be sent to this address as a high priority mail. Multiple addresses can be separated by a comma. If this field is left blank or the configuration parameter is mission, no E-Mails will be sent by default. Recipients entered on a per connection basis will still receive notifications.

Send Log

Check this box to attach a copy of the log file to the email notification. Log Files generated at default level 2-Lowest should send without issue, but be aware that logs produced at level 3- Detailed may be too large for most mail systems.

Email Notification box

This box on the Email Notify tab of each connection allows you to fine tune recipients per connection. Recipients specified via Global SMTP will receive ALL notifications, but if you want to specifically send only certain notifications per connection, enter email addresses here, on the Email Notify tab.

Back-up and Restore of UnitySync

Creating a back-up of UnitySync is just like backing-up a directory. To back-up, simply zip and/or copy all of the UnitySync files including programs, custom map files, and connection found in ...\\UnitySync to an another location.

To restore UnitySync, copy the entire ...\\UnitySync back-up directory and paste to the original location or to a new server. Also, an individual connection may be copied to restore a specific connection.

Once you have backed-up UnitySync, you may delete the data .db file of each connection to save space. Please note that if you used these backed up connections for future syncs, if there are any discrepancies between the Source system and the hash table, they will work themselves out in a few synchronization runs.

You may need a new license key. If you receive a license error, contact keys@dirwiz.com for a new key.

You will need to reconfigure your IIS virtual directory. Please refer to the white paper Web Server Configuration to refresh your memory.

Move or Reinstall UnitySync

Each UnitySync installation is licensed to run on a specific server identified by the serial number found on the UnitySync landing page.

Licensing rules apply when reinstalling or moving UnitySync to another server. The Serial Number is unique for every installation, so you'll need a new license key. To obtain one, contact keys@dirwiz.com. Let us know the original licensed serial number (prior to reinstalling, upgrading or moving) and the new serial number as it appears on the UnitySync launch page. See the licensing section for more information.

Reinstalling a previous version is only possible if you have preserved your original installation files. UnitySync cannot distribute installation files for versions previous to the one currently available on our web site Downloads page.

Changing hostname or installation directory

You can **move** the files as noted in Method #1 below. You will need a new license key (see above).

Moving your UnitySync installation

In order to move the UnitySync installation to a new server, you will have to transfer the program files and connection information to the new server, then obtain a new license key. This can be accomplished in one of two ways.

Prior to the Move:

Tip 1: it is highly recommended that you run one last Discovery and Synchronization from the currently installed location.

Tip 2: If your install has a large backlog of log files, consider running Clean Logs.

Move or Reinstall Options

Method #1:

Transplant \UnitySync directory to new server

1. **Before you start your move**, please review the new System Requirements for UnitySync. This includes system requirements as well as *special requirements related to o365 connections*.
2. Simply copy the entire \UnitySync directory from the original location to the new server. This brings over all needed files from the previous version.
3. If you automate your connections by running a script via a scheduled task, copy the script over to the new server.
4. Make note of your Scheduled Task settings as you'll want to set up similar Tasks on the new server.

Once you complete Method #1 steps, refer below to "Re-enable connections after reinstall/move."

Method #2:

Re-install and move config/custom files

This is the recommended procedure if you are **upgrading** UnitySync software at the same time you are moving to a new server.

1. Got to UnitySync Upgrade Notes and Requirements. Follow all instructions to confirm requirements and complete upgrade instructions are provided here.

Final step (for either method) to Re-enable connections after reinstall/move

After using either of the above methods you must continue to the following steps in order to re enable your connections and scheduled tasks.

1. **Reset file permissions on install directory:** When \UnitySync or its subfolders are copied to a new server/location, the file permissions may be lost. It may be necessary to set file permissions as noted in this KB article.
2. **Request a new license key:** Licensing rules apply when you are moving an installation of UnitySync to another server. You will need to request a replacement license key. Please see Requesting a Permanent License key for detailed instructions to obtain a new key.
3. **Test:** Before running a sync on this new install, we recommend you test using Simulation. Review the SIM log for any unexpected Add/Mod/Delete actions.
IMPORTANT NOTE: *It is not uncommon for a version upgrade to trigger mods to all objects, so don't be too concerned if that is what you see. If you have any questions about your Simulation results, contact Technical Support.*
4. **Reset IIS virtual directory:** If you are using a Web Server rather than dirweb.exe, you will will need to update settings for the UnitySync virtual directory. Refer to the Web Server Configuration document for more information.
5. **Resuming Scheduled Syncs:** You are probably running your connections automatically via use of scheduled batch scripts. Be sure you have copied your script from the old server to the new. If necessary, edit the script on this new server, such as to identify the correct install path of UnitySync (if now different). Recreate your Windows Scheduled Task settings on the new server to automate your connections as desired.

This completes the steps to Move or Move/Upgrade your UnitySync software on a new server. Please contact Technical Support for additional help if necessary.

Uninstall UnitySync

There is no Uninstall program. You can simply delete your ...\\UnitySync directory. We suggest you backup the directory first, in case you want to reinstall later. There are no registry entries to clean up. You will want to remove the UnitySync IIS virtual directory.

Troubleshooting

FAQ

Visit the Troubleshooting topic of the knowledge base for help with most questions.

Troubleshooting: Log Files

UnitySync creates a log file for each synchronization run. Discovery, Simulation, and Synchronization phases generate their own log files. The files contain five levels of error logging, 1 through 5, with 1 the lowest and 5 the highest.

Most situations only call for the default error logging level of 2. The log files are in .txt format and are viewable in UnitySync or in a text editor like Notepad. Larger log files (generated at Log File level 3-Detailed or higher) may need to be viewed in a text viewer equipped for larger files.

Log files are located in ...\\UnitySync\\Connections\\YourConnectionName\\Logs.

To view the log files, you may open them from this location or select the file from the drop-down in the Logs section of the General tab, then click the Download Log button.

If low level logging is chosen (2-Lowest), the first synchronization run will be the largest and will contain several lines of information for each object that is synchronized. Later synchronization runs will only log changes (deltas). We recommend that you set your logging level to 2-Lowest as a default.

Higher level logging (3-Detailed, 4-Verbose, 5-Debug) generates a more detailed log. These detailed logs may be very large, so only enable high level logging to troubleshoot issues. For most issues, a log generated at 3-Detailed is sufficient for troubleshooting.

If you configure the global E-mail Notifications option in UnitySync, it will send a summary email for each successful synchronization run, or an email when an error occurs. Choose whether or not you want the log file attached to these messages.

Interpreting Log Files

Looking at a discovery or sync log file, you'll see that the top part of the file has several sections. These sections are the steps UnitySync takes to prepare for a synchronization run.

- The configuration parameters that were defined through the Admin program are loaded.
- The map, structure, and exclusions files are loaded.
- The Source and Destination directories are discovered and accessed.
- The software license is detected before any input/output takes place.

The Discovery log shows the search of each Source container, all excludes and all objects successfully exported. The Run Summary will show the total number of objects processed and if there were any errors. The Sync log shows additions, modifications and deletions (deltas) made to containers and user objects. The Run Summary shows the total number of additions, modifications, deletions (deltas) and entries processed. It will also show any errors, if there were any.

Take a look at the logs to make sure that you have the number or entries processed that you expected to have. Remember, after the initial sync is done, only changes that have made Source entries different from Destinations entries will be processed.

UnitySync log files are important to troubleshooting most synchronization errors. Email the connection's log file(s) to Technical Support when you ask for help from tech support.

Find the log files in: ...\\UnitySync\\Connections\\YourConnectionName\\Logs

There are up to three log files (discovery, sim, sync) for each synchronization run. All logs for your specific connection should be sent to support unless otherwise requested. Zip the log files, or upload large zipped log files to our secure upload site.

Deleting Log Files

UnitySync names the log files based on date and time and are not automatically deleted. Delete them manually or by configuring `--cleanlogs` as described in Log File Management.

Troubleshooting: The SQLite Database and Bad Data Files

UnitySync maintains data files on synchronized objects for each connection in a SQLite database. The database is built with the very first synchronization run and contains a reference (by DN) to each synchronized object, linking Source to Destination. Every synchronization run includes an integrity check of the database. If a bad database is detected, the synchronization run creates an error indicating a bad data file. Should this happen, first try the ForceMods command.

If ForceMods is not sufficient to rectify your issue, the Synchronization Recovery Procedure may be necessary. Make sure to review all instructions and cautions before performing this procedure. Detailed instructions for the Recovery Procedure are found in the referenced article.

Troubleshooting: Fail and Restart for an interrupted Sync

An interrupted synchronization run can be restarted irrespective of why it was interrupted. UnitySync will restart from the beginning and run normally. Any processing that happened before the interruption will be noted in the log with an error message, but for informational purposes (e.g. deleting an entry that was already deleted).

Troubleshooting: Contacting Support

Information on options for contacting Technical Support are on our website.

Phone support is most efficient and productive if you first forward relevant log files and/or error messages, with a brief synopsis of your issue, prior scheduling a call. Additionally, many issues are easily resolved via email, providing you with detailed instructions for future reference.

For information on forwarding log files to support for assistance, please review this knowledge base article.

Appendix

The following appendices contain expanded information on:

- Trim DN
- Custom Mapping
- Filters & Exclusions
- Automating your UnitySync Connections

Appendix 1: Trim DN

Trim DN

Trim DN attribute is found on the Destination tab and is generally used in conjunction with levels greater than zero.

Positive Trim DN Values

Positive Trim will remove DN components from the left, omitting the component from the Destination structure when the new object is created. This feature may also be used to add components that might not otherwise be included in the default structure replication. Only Source domain components included in the 'Source Context' may be trimmed using this feature.

IMPORTANT NOTE: *The actual components available for trimming as well as the actual Trim results will vary depending on your environment and connection configuration. The best way determine if Trim DN is useful in your connection is to run Discovery and Simulation. Then review the log file to review the container structure of each object as it is added.*

Any number of level trimming is supported, 1, 2, 3, etc. Any level above the number of Source Context structures will be ignored as Source container structure is never trimmed.

Positive Trim DN Examples

The following examples assume the following:

- **Source IP** is a domain controller (but not a global catalog). If the Source IP points to a domain controller, then Source Context may be simply dc=com and therefore only dc=com is available for trimming.
- **Source DN:** In format of cn=user,ou=sub2,ou=sub1,ou=contacts,dc=domainA,dc=com
- **Struct Name:** Your sync container/ou name; example SyncContainer
- **Source Context:** In format of dc=domainA,dc=com
- **Selection DN** is not specified.
- **Levels:** All

Examples

In this example, only 2 levels are available for trimming. So any higher trim will simply return the same results as Trim=2.

Trim DN=0

Destination DN: cn=user,ou=sub2,ou=sub1,ou=contacts,ou=domainA,ou=com,ou=SyncContainer,dc=DestDomain,dc=com

Trim DN = 1

Destination DN: cn=user,ou=sub2,ou=sub1,ou=contacts,ou=domainA,ou=SyncContainer,dc=DestDomain,dc=com

Trim DN = 2 (default / blank) Destination DN: cn=user,ou=sub2,ou=sub1,ou=contacts,ou=SyncContainer,dc=DestDomain,dc=com

Trim DN = 3 and up Destination DN: cn=user,ou=sub2,ou=sub1,ou=contacts,ou=SyncContainer,dc=DestDomain,dc=com

Negative Trim DN values

Negative Trim values may be used when a container or domain is specified in the Source Tab's Selection DN parameter. Negative Trim values always add DN components that are not part of the replicated structure by default.

Like Positive Trim, Negative Trim function only Adds structure based on the Source Context specified. In this case, the Source Context is equal to the Selection DN.

Negative Trim DN Examples

The following examples assume the following:

- **Source DN:** In format of cn=user,ou=sub2,ou=sub1,ou=contacts,dc=domainA,dc=com
- **Struct Name:** Your sync container/ou name; example SyncContainer
- **Source Context:** In format of dc=domainA,dc=com
- **Selection DN** is ou=sub1,ou=contacts,dc=domainA,dc=com
- **Levels:** All

In this case we are pointing directly to ou=sub1 using Selection DN. Normally, this level and any above it would not be replicated to the Destination. Negative Trim allows you to opt for these levels to be added back in from left to right.

Trim DN = blank (default) Destination DN: cn=user,ou=sub2,ou=SyncContainer,dc=DestDomain,dc=com

Trim DN = -1 Destination DN: cn=user,ou=sub2,ou=sub1,ou=SyncContainer,dc=DestDomain,dc=com

Trim DN = -2 Destination DN: cn=user,ou=sub2,ou=sub1,ou=contacts,ou=SyncContainer,dc=DestDomain,dc=com

Trim DN = -3 Destination DN: cn=user,ou=sub2,ou=sub1,ou=contacts,ou=domainA,ou=SyncContainer,dc=DestDomain,dc=com

If you have any questions regarding Trim DN and its uses. please contact Technical Support.

Appendix 2: Custom Mapping

Custom Mapping

There may be times when you need to customize the attributes of your synced objects and attributes. A typical map file is based on assignments. The value on the left of the expression identifies the Destination attribute. In the case of an LDAP Destination, this should be the LDAP attribute name where you want to put the incoming value. The value on the right of the expression identifies the Source attribute to pull the value from. In the following example, we will assign the Destination "telephone number" attribute the value of the Source "pager number" attribute.

```
telephonenumber=^pager number^
```

To do this, you must create your own Custom File and include it in the configuration of a connection. It is recommended that you contact Technical Support if you are unfamiliar with modifying a custom map file. It is important that you understand how these mappings are designed to work with the UnitySync programs before using this feature. Equally important is an understanding of your directory schema. In most cases, you will only need to modify the Object (person) Map File.

To create a Custom file for any of the available map files, click the relevant map file type button. Object Maps and Sourcedefs are the two most commonly used custom files.

Creating a Custom Object Map or List Map File

A Custom Object Map file allows you to change any aspect of how the sync creates new objects on the Destination directory for this connection. Object Map Files are directory type specific. The Source and Destination directory type of the connection must be appropriate for a specifically configured Object Map file. For instance, a Notes-to-AD map file may not be used in an AD-to-Notes connection.

The most common use for a Custom Object Map is to customize how attributes are mapped between systems.

Custom List Maps are very similar to Custom Object Maps but they apply only to synced Groups. If you are syncing Groups as email addresses (default option), the sync will use your Object Map (default or custom) for syncing. If, however, you have created a Custom List Map, it will use that map instead.

If you are syncing Groups as Groups, the sync will use your List Map (default or custom) for syncing.

Custom Mapping: Creating a Custom Object Map file

A Custom Object Map file allows you to change any aspect of how the sync creates new objects on the Destination directory for this connection. Object Map Files are directory type specific. The Source and Destination directory type of the connection must be appropriate for a specifically configured Object Map file. For instance, an AD-to-AD map file may not be used in an Notes-to-AD connection.

The most common use for a Custom Object Map is to customize how attributes are mapped between systems. For more help on Custom mapping syntax, refer to the Custom Mapping: Mapping Syntax.

Creating a Custom Map File in UnitySync

1. From the UnitySync Manager User Interface (UI), select your connection, then click the Custom tab.
 - Click the Object Map button.
 - If you already have a custom map file, it will open. If not, a popup will prompt you to provide a name to create one. Enter a name in the Object Map parameter, then click Create.
 - The map file with default mappings will open.
 - Make the desired changes to the map file. (Note: The DN and objectclass mappings must remain the **top two** mappings).
 - Click Save to save your map file changes.
 - Click Save again at the bottom of the Custom tab to save your selection of the custom map for this connection.

Custom Mapping: Creating a Custom List Map file

A custom list map file allows you to change any aspect of how the sync creates new **Group** objects on the Destination directory for this connection. Default mappings for list map files differ according to directory type. The Source and Destination directory type of the connection must be appropriate for any given list map file. For example, a Notes-to-AD list map file may not be used in an AD-to-AD connection.

The most common use for a custom list map is to change how attributes are mapped between systems. It is also possible to customize the sync container, so that your synced group objects are created in a different container than synced contact or user objects.

IMPORTANT NOTE: *A custom list map is typically not needed if you are syncing Groups as Contacts (default group sync functionality). When you sync Groups as Contacts, the object map will be used by default. If you are syncing Groups as Contacts and feel you need to alter the map file, please contact our technical support team to discuss your needs.*

Creating a Custom List Map File

1. In the UnitySync UI, select your connection, then click the Custom tab.
2. Click the List Map button, then type in a name for your new custom list map and click Save.
3. The map file will open with default values. Make any desired changes to the map file.
4. Click Save to save your map file changes.
5. Click Save again at the bottom of the Custom tab to save your selection of the custom list map file for your connection.

A Note about using a custom DN mapping to change your sync container for Groups:

The `~struct~` is a required part of the syntax. This value will be the structure specified by PlacementDN and/or StructName.

You may **not** remove `~struct~` and try to hard code the entire DN mapping, such as:

```
dn=cn=~component0~,ou=Groups,dc=domain,dc=test
```

This syntax will result in inaccurate structure.

You **may**, however, customize the DN to add structure where `~struct~` is a common root of the destination sync container for both Contacts and Groups.

For example:

Placement DN is: `dc=domain,dc=test`

Custom Contact (Object) DN mapping: `dn=cn=~mail#64~,ou=Contacts,~struct~`

Custom Group (List) DN mapping is: `dn=cn=~component0~,ou=Groups,~struct~`

With these mappings:

Your Groups will be synced to: `ou=Groups,dc=domain,dc=test`

Your Contacts will be synced to: `ou=Contacts,dc=domain,dc=test`

For additional guidance on custom mapping syntax, refer to the UnitySync Administrator's Guide. Please contact support@dirwiz.com for assistance, if needed.

Custom Mapping: Creating a Custom Sourcedef file

A custom Sourcedef file allows you to change certain aspects of how the sync will query the source LDAP directory for this connection. Sourcedef files are directory type specific; i.e., an Active Directory (AD) Sourcedef is not interchangeable with a Notes Sourcedef.

Modifications to the Sourcedef should be done carefully as a misconfigured Sourcedef may break your connection. It is recommended you contact Technical Support for assistance.

The most common uses for a Custom Sourcedef file:

1) Customize the **attribs=** parameter to add custom/expanded attributes (or remove attributes).

attribs= defines all attributes that will be pulled from the Source directory for this connection.

2) Customize person-objectclass parameter. This allows you to assign multiple objectclass to be included on Sync. (and to resolve Unknown Source ObjectClass warnings).

How to Create a Custom Sourcedef File

- From the UnitySync Sync Manager User Interface (UI), select your connection, then click the Custom tab.
- Enter a name in the Sourcedef parameter, then click Create.
- The Sourcedef with default mappings will pop up. Make the necessary changes to the Sourcedef file. Contact support@dirwiz.com if you need assistance.
- Click Save to save your sourcedef file changes.
- Click Save again at the bottom of the Custom tab (to save the custom sourcedef for this connection).

IMPORTANT NOTE: Do not select an existing custom Sourcedef file from the drop down prior to clicking the Sourcedef button unless you are SURE you want to edit the existing custom map file.

Once customizations are complete, ALWAYS run a Discovery and SIMULATION to verify results before running a sync.

Custom Mapping: Creating Custom Folder or Struct Map Files

These options are not commonly used. If you believe you need to utilize custom mapping for folder or structure, please contact our Technical Support Team for assistance.

Custom Mapping: Mapping Syntax

It is important to use correct syntax when building your own map file. Below find descriptions of the various control characters:

Character	Example	Description
^	^givenname^	This represents the first value of a Source attribute.
[]	[objectclass]	This represents all values of the enclosed attribute. This is handy when assigning all values of an attribute in one assignment. (Not used in ODBC)
~	~sn~	This represents a UnitySync generated internal variable.
\		top\
{ }	{telephone}	This performs a 'pop' function on the telephone attribute. The first value is assigned and removed from the Source variable. The next assignment by ^ or { } will be the next value of telephone. This option is not used in ODBC.
r\		r\

Using Reserved Characters as Literals

Some characters in map files are reserved for special use by UnitySync. If you need these characters to be included literally in a custom mapping, you must 'escape' the character with a preceding \ (backslash) character. The following list identifies the escape sequence for each of the reserved characters:

Character	Example
\]]
\[[
\{	{
\}	}

Character	Example
\~	~
\^	^
\#	#
\%	%

For example, the following mapping appends the company attribute value surrounded by square brackets to the displayname:

```
displayname#256=~cn~ \[^company^\]
```

This mapping would result in displayname values as follows:

- Oprah Winfrey [Harpo Productions]
- Bill Gates [Microsoft]
- Warren Buffet [Berkshire Hathaway]

LDAP Mapping Examples

Source attributes:

```
1 dn:uid=jsmith,ou=ABC Company
2 objectclass:top
3 objectclass:person
4 telephone:555-1212
5 telephone:653-2234
6 mail:jsmith@abc.com
7 company:ABC Company
8 cn:Smith, John
9 address1:10 River Street
10 address2:Suite 100
```

Attribute Mappings:

```
1 dn=cn=~cn~,~struct~
2-3 objectclass=interorgperson|top
4 tel1={telephone}
5 tel2=^telephone^
6 mail=^mail^
7 company=Acme
8 last=~sn~ and first=~givenname~
9-10 r|address=^address1^|^address2^
```

IMPORTANT NOTE: *telephone and objectclass are multivalued fields.

How are the results determined?

In our configuration we have set display-parse=yes and display-firstlast=yes on the Display Name tab.

- **Line 1:** Since the Source CN (Line 8 of the LDIF) is parsed using display-parse=yes and displayfirstlast= yes the internal variable of ~cn~ will be constructed. The ~struct~ variable autogenerates the rest of the dn string and builds any structure that needs to be built.
- **Lines 2&3:** The Destination object class has two literal values 'inetorgperson' and 'top'.
- **Line 4:** Because the Source telephone field (Lines 4 & 5) have 2 values, the first value will have to be popped off. The {telephone} variable pulls off the value '555-1212' and assigns it to tel1. What is left in the Source telephone field is one value '653-2234'.
- **Line 5:** tel2 is assigned what's left of the telephone field: '653-2234'.
- **Line 6:** This is a straight attribute assignment.
- **Line 7:** The Destination company attribute will be assigned 'Acme' for all objects.

- **Line 8:** Because display-parse=yes the variable ~sn~ will be the result of the parsed CN (Source line 8). If the display-parse was set to no, ~sn~ would have been assigned Source attribute sn if it were available.
- **Lines 9&10:** The preceding r\ indicates that the following " \ " symbols will be translate to CR-LF. Therefore, the value of the Destination attribute "street address" will be:
10 River Street
Suite 100

So, our result looks as follows in the LDIF file:

```
1 dn:cn=John Smith,o=Acme
2 objectclass:inetorgperson
3 objectclass:top
4 tel1:555-1212
5 tel2:653-2234
6 mail:jsmith@abc.com
7 company:Acme
8 last:Smith and first:John
9&10 Address::MjAgU29tZSBTdHJlZXQNC1N1aXRlIDewMA==
```

IMPORTANT NOTE: Multi-line attributes contain control characters (CR-LF). Therefore, LDIF standard requires that the attribute be base64 encoded, as show in Lines 9&10 above.

Custom Mapping: Destination DN

The mapping for the Destination DN **must** be the first line in your custom map file. This mapping defines the Destination DN that will be built for each synced object.

For example, the default mapping to create contact objects on an Active Directory Destination generally reads as follows:

```
dn=cn=~mail~,~struct~
```

CN represents the primary mail attribute of the Source object. This is the preferred CN value because it is very likely to be unique.

IMPORTANT NOTE: *Unless your needs strictly require it, it is not recommended that you modify the default DN mapping.

Whatever DN mapping is specified, the Source value used as the CN may sometimes include characters that are invalid for a Destination DN. UnitySync will automatically strip the following characters if they appear in the Destination DN:

```
% ~ & / # +
```

For more information on customizing your DN mapping for syncs to Active Directory or Notes please click through to the relevant knowledge base articles.

Custom Mapping: Add SMTP/smtp proxyaddresses at Sync time

Syncing to an Active Directory (AD) Destination, the default mapping for proxyaddresses (shown below) will sync all Source SMTP/smtp addresses. The corresponding contact created on the Destination will contain all these same addresses. **This default is usually all that is needed for a typical directory sync.**

```
proxyaddresses=[proxyaddresses]
```

We do recommend that you use Address Policies whenever possible as they offer more flexibility in the address format being created. Though these address policies are typically enabled, it's possible you may need to enable them at sync time. Instructions for this are available in our knowledge base.

Occasionally, there might be the need to add an additional **single** smtp address on the Destination contact.

- Leave the default mapping in place.
- Add the below mapping. This will take the primary smtpname and append the domain name specified to add a single smtp address.


```
proxyaddresses=smtp:~smtpname~@NewDomain.com
```

Rarely, there may be a need to add another smtp address (with the same domain) based on ALL Source smtp addresses. This is possible via a custom mapping that calls an eval.js javascript sub routine.

Example Scenario:

Source values are:

- proxyaddresses: SMTP:maryjanemmm@domain.com
- proxyaddresses: smtp:maryjanesmith@other.com

Desired Destination values are:

- proxyaddresses: SMTP:maryjanemmm@domain.com
- proxyaddresses: smtp:maryjanesmith@other.com
- proxyaddresses: smtp:maryjanemmm@TheNewDomain.com
- proxyaddresses: smtp:maryjanesmith@TheNewDomain.com

IMPORTANT NOTE: *The first two sync directly from the Source, while the second two are created at sync time from the Source values and the desired domain.*

The solution for this includes a custom mapping which calls a javascript function:

1. In a custom object map file, leave the default mapping as-is
2. Add a new line with this additional mapping for proxyaddresses:

```
proxyaddresses=smtp:&grabproxy('[allmail]')&@TheNewDomain.com
```

3. Per this article, edit the eval.js file to add this sub routine:

```
<CR> function grabproxy(proxies) { names = proxies.split("@"); return names[0];
}
```

Custom Mapping: Syncing non-smtp proxyaddresses

The default mappings for proxyaddresses only brings over the SMTP/smtp addresses.

```
proxyaddresses=[proxyaddresses]
```

By adding additional mappings as described below, you can bring over other proxyaddress types. The three mappings will bring over all X400, X500 and SIP addresses. We refer to these as 'mail component mappings', with the components being each individual non-SMTP mail type (i.e. X500, X400, SIP, etc).

First, if you don't already have a custom object map file, create one. Then, add one or more of these mappings in addition to the default mapping shown above:

```
proxyaddresses=x400:[proxyaddresses-x400]
proxyaddresses=x500:[proxyaddresses-x500]
proxyaddresses=sip:[proxyaddresses-sip]
```

If you do not get the results you desire, please contact support@dirwiz.com for assistance.

Custom Mapping: String Manipulation

String manipulation is used within custom map files to truncate or extract characters from a string. If you have a unique format for an attribute we suggest you create a custom map file and use the examples below as a guide to producing desired results. Of course, you are always welcome to contact our Technical Support Team for assistance.

The basic syntax for string extraction is:

```
[dest attribute]#[Num of characters],[Offset]=^some source attrib^
```

Please note, [Num of characters] refers to how many characters you would like to extract from the string, and may be specified alone. [Offset] refers to where in the string to begin pulling from. Zero or no value means to

start at the left most character. A positive value means so many characters from the left. A negative value means so many characters from the right. When using [Offset], [Num of Characters] is required.

Examples

In this case we will manipulate the title attribute. The source value for this attribute is *president*.

Pull 4 characters from the left

```
title#4=^title^ or title#4,0=^title^
result: pres
```

Pull 4 characters from the right

```
title#4,-4=^title^
result: dent
```

Pull the 5th character

```
title#1,4=^title^
result: i
```

Pull the last character

```
title#1,-1=^title^
result: t
```

Remove last 5 characters

```
title#-5,0=^title^
result: pres
```

The above examples will work on all versions of UnitySync.

If you want to build a value from multiple source attribute values, however, you may use scripting in combination with string manipulation. A mapped value may be constructed from multiple source fields or with a combination of fixed values. For example, if you want the first letter of given name, period, full last name, dash, last 4 digits of phone number...

IMPORTANT NOTE: In UnitySync v2.4 and later use javascript in combination with string manipulation. In versions earlier than v2.4, we urge you to upgrade to the latest version.

Example 1

In this example, we will concatenate and trim data. Our Source data includes:

```
telephonenumber=5554563239
sn=Smith
givenname=Jane
```

If we map:

```
extensionattributel=^givenname#1^.^sn^ -&'^telephonenumber^'.slice(-4);&
```

```
'homephone=(^telephonenumber#3^) &'^telephonenumber^'.slice(3,6);&-&'^phtelephonenumberne^'.slice(
```

The Synced values will be:

```
extensionAttributel=Jane.Smith-3239
homephone=(555) 456-3239
```

Example 2

In this example, we will reformat the date. Our Source data is:

```
whenCreated:20141210135109.0Z
```

And we map:

```
extensionAttribute1=&'^whencreated^'.slice(0,4)&:&'^whencreated^'.slice(4,6)&:&'^whencreated^'.slice(6,10)
```

The Synced value will be:

```
extensionAttribute1=2014:12:10T13:51:09
```

Please see the Javascript category in our knowledge base for more information on scripting. For additional assistance with these types of mappings, please contact our Technical Support Team.

Custom Mapping: Javascript Mapping

Attribute mapping can be enhanced by the use of Javascript. This includes scripting within your custom map files as well as via eval.js subroutines.

Below find examples of commonly used javascript mappings.

To insert country code to a telephone number

The below mapping will insert a country calling code if the Source data is missing it. In this example, we are using +44 for the UK.

```
NewPhone=&"^telephonenumber^".substr(0, 1) == '+' \? '^telephonenumber' : '+44 ^telephonenumber'
===
```

Slice source data for specific format

The below mapping will:

- Insert prefix 'Emp'
- Grab the final 4 characters of Empnr
- Drop the / from HireDate
- Returns result as one string

Source data:

```
Empnr=001234
HireDate=03/10/2018
```

Mapping:

```
extensionAttribute3=Emp+"&"^Empnr^".slice(-4)&&"^HireDate^".replace(/\//g, '')&
```

Return value and format:

```
Emp123403102018
```

```
===
```

More Sample mappings:

Convert a string (number) to a an integer so you can set an integer only dest attribute (i.e. enabledprotocols).

```
enabledprotocols=&parseInt("^department^")&
```

Set destination conditionally based on source attribute value).

```
company=&'^company^' == " \? 'None Specified' : '^company^'&
```

```
===
```

For more complex javascript commands, refer to Sample Javascript Functions. Using eval.js with custom mappings allows for more complex data manipulation.

Custom Mapping: Javascript functions for eval.js

The details below pertain only to Javascript for use with **UnitySync v2.4** or later. If you are using a version prior to v2.4, review this article for additional information.

In ...\\UnitySync\\-v2.4\\global, determine if you already have an eval.js file. If you do not, you will find the sample file eval-example.js Copy this sample file to eval.js, then right-click to open and edit your eval.js. This contains one sample subroutine.

Below are additional sample subroutines and the appropriate custom mapping syntax to call them.

smtp_domain

This function takes a string as input (email address) and returns the value after the @ character.

```
function smtp_domain(smtpaddr)
{
    var parts=smtpaddr.split('@');
    return parts[1]==undefined ? '' : parts[1];
}
```

Example mapping to call smtp_domain: Attribute=&smtp_domain('^mail^')&

swapchars

This function will convert dash and space to period and 'contr' to blank

```
function swapchars(mystring)
{
    var str = mystring.replace(/-/g, '.');
    str = str.replace(/ /g, '.');
    str = str.replace(/Contr/g, '');
    return str;
}
```

Example mapping to call swapchars: description=&swapchars('^description^');&

onlydigits

This function will return a value containing only digits.

```
function onlydigits(mystring)
{
    var str = mystring.replace(/\D/g, '');
    return str;
}
```

Example mapping to call onlydigits: department=&onlydigits('^department^')&

NOTE: This may also be accomplished without eval.js. Instead use only this single custom mapping:

department=&'^department^'.replace(/\D/g, '');&

titlecase

This function will return the string using Title Case.

```
function titlecase(str)
{
    fixme=str.replace(/\w\S*/g, function(txt){return txt.charAt(0).toUpperCase() + txt.substr(1).toLowerCase();});
    return fixme;
}
```

```
}
```

Example mapping to call titlecase: `description=&titlecase('^description^');&`

remove_space

This function will strip all spaces from the attribute.

```
function remove_space(str)
{
return str.replace(/\s/g, '');
}
```

Example mapping to call remove_space: `sn=&remove_space('~sn~')&`

NOTE: *This may also be accomplished without eval.js. Instead use only this single custom mapping: `sn=&'~sn~'.replace`*

remove_space_and_apostrophe

This function will strip all spaces AND apostrophe from the attribute.

```
function remove_space_and_apostrophe(str)
{
return str.replace(/\s/g, '');
}
```

Example mapping to call remove_space_and_apostrophe: `sn=&remove_space_and_apostrophe('~sn~')&`

NOTE: *This may also be accomplished without eval.js. Instead use only this single custom mapping (Note the use of double and single quote in this mapping): `sn=&"~sn~".replace(/\s/g, "").replace(/'/g, "");&`*

fix_string

This function will return the string after removing or swapping the specified characters. In this example, the various characters are replaced with a blank (effectively removing that character).

```
function fix_string(orig)
{
var newstr=orig.replace(/#/g, '');
newstr=newstr.replace(/-/g, '');
newstr=newstr.replace(/'/g, '');
newstr=newstr.replace(/~/g, '');
newstr=newstr.replace(/\\/g, '');
return newstr;
}
```

Example mapping to fix string: `description=&fix_string('^description^')&`

grab_container

This function accepts a single text value (i.e. DistinguishedName or DN) and removes everything before the first instance of "OU=". The remainder of the string is returned. If "OU=" is not found, the entire string is returned.

```
function grab_container(fullldn)
{
var parts=fullldn.substring(fullldn.indexOf('OU='));
return parts;
}
```

Example mapping to call grab_container: `extensionAttribute1=&grab_container('^distinguishedName^')&`

If this is the DN of the source object:

```
CN=enichols,OU=My Users,DC=2k10,DC=test
```

Following Sync, the Destination extensionAttribute1 will be:

```
OU=My Users,DC=2k10,DC=test
```

IfThenElseReturn

This function will accept multiple values, perform if/then/else to determine desired output value and return that value. Can be modified for more input parameters and/or text values.

example mapping:

```
description#1024=&IfThenElseReturn('^department^','^office^','^l^','^c^')&
```

```
function IfThenElseReturn(var1,var2,var3,var4)
{
    if (var1 == "1001") {
        return var2 + '- internal';
    } else if (var1 == "1002") {
        return var3 + '- local';
    } else if (var1 == "1003") {
        return var4 + '- international';
    } else {
        return var1 + '- unknown';
    }
}
```

NOTE: Functions may also be called without parameters. This would require the use of getsrc and getint.**

i.e. Description=&return_two_values_not_passed()&

ReturnYMD

This function will take as input an AD Date string (i.e. AccountExpires) and return YYYY/MM/DD

// Input: accountexpires 100-nanosecond intervals since 1601/01/01 UTC

// Return: YYYY/MM/DD or "6"

```
function ReturnYMD(s)
{
    if (s == 9223372036854775807) { //default, not set, Return "6"
        return "6";
    } else if (s == 0) { //Never Expire, Return "6"
        return "6";
    } //Else...Continue and return YYYY/MM/DD
    var t=((s/10000000) - 11644473600);
    var d=new Date(0);
    d.setUTCSeconds(t);
    var x = d.toISOString();

    var yrmoday = x.slice(0, 10); //grab first 10 digits YYY-MM-DD
    yrmoday = yrmoday.replace(/-/g, '/'); // replace char - with char /

    return yrmoday; // Return YYYY/MM/DD
}
```

Example mapping to call ReturnYMD:

```
description=&ReturnYMD("^accountexpires^");&
```

formatDisplayName

Note: This function utilizes `getsrc` instead of passing in parameters from the original mapping.

```
function formatDisplayName()
{
  var sn = getsrc('sn');
  var gn = getsrc('givenname');
  var nn = getsrc('nickname');
  var ecd = getsrc('exportcontroldata');
  var mn = getsrc('middlename');
  var name = '';

  if (typeof(sn) != "undefined") {name = sn;}
  if (typeof(nn) != "undefined") {name = name + ', ' + nn;}
  else if (typeof(gn) != "undefined") {name = name + ', ' + gn;}

  if ((typeof(nn) == "undefined") && (typeof(mn) != "undefined" )) {name = name + ' ' + mn.substr(0, 10);}

  if (ecd == 'allowed') {name = name + '      Allowed'}
  else {name = name + '      Export License Required US'}

  return name;
}
```

Example mapping to call `formatDisplayName`

```
displayname#256=&formatDisplayName() &
```

ParseDisplayName

Takes as input a string (`displayname`, presumed format "First.Last"). Returns `displayname` in format "Last, First". If NO period (.) in string, returns original string.

```
function ParseDisplayName(origdisplayaname)
{
  var parts=origdisplayaname.split('.');

  if (parts[1]==undefined) return origdisplayaname;
  return parts[1] + ', ' + parts[0];
}
```

Example mapping to call `ParseDisplayName`

```
Displayname=&ParseDisplayName("^displayname^") &
```

Debugging javascript

For more information on debugging, please see [Debug logging in custom eval.js functions](#)

Custom Mapping: Setting legacyExchangeDN on synced AD contacts

Typically, the the use of a destination Domain Admin login ID returns the valid `legacyexchangedn` root value. However, we have seen instances of an error even when using Domain Admin.

The error looks like this:

```
Error Summary : 8/1/2011 12:38:24 PM
LegacyExchangeDN Failure | 1
Fatal Error | 1
```

This error indicates the auto detection of destination legacyexchangedn was **not** successful. In order to proceed, you must manually identify the destination legacyexchangedn root value. Then, insert the value using the override parameter 'exch-legdn' in config.txt.

Note: If appropriate, there is a way to disable the auto detection of the destination legacyexchangedn root value. See the Method #3 below.

There are two ways to find the value for the destination legacyexchangedn root:

Method #1

To identify the appropriate override value we suggest using ADSI Edit to view the legacyExchangeDN property.

- From the root of your domain, select:
 - CN=Configuration,
 - then CN=Services,
 - then CN=Microsoft Exchange,
 - then CN=YourExchangeOrg,
 - then CN=Administrative Groups.
- Now look for the appropriate Administrative Group for your Exchange version:
 - for Exchange 2003
 - * CN=First Administrative Group
 - for Exchange 2007
 - * CN=Exchange Administrative Group (FYDIBOHF23SPDLT)
 - for Exchange 2010
 - * CN=Exchange Administrative Group (FYDIBOHF23SPDLT)
- Then click Properties.
 - Look for the Attribute 'legacyExchangeDN'.
 - Click Edit to view the value. Click Cntl-C to copy the value.

/o=E2k Organization/ou=First Administrative Group

IMPORTANT NOTE: This may be called something different in your organization.

Method #2

Alternatively, you may find the legacyExchangeDN attribute value of a contact that was manually created on the Destination. (You can see this in the contact's attribute properties via Users & Computers).

For example:

/o=E2k Organization/ou=First Administrative Group/cn=Recipients/cn=John.Smith

IMPORTANT NOTE: You do **not** need to include the '/cn=Recipients/cn=name' portion of this value as it is appended by the default mapping.

Using the exch-legdn override in config.txt

Once you know the proper legdn root value, add it to your configuration as follows:

- Go to the Custom tab in UnitySync
- Click the Raw Config button.
- Add a line with the value in the following format:

exch-legdn=/o=org/ou=YourAdministrativeOU

Once exch-legdn is set, leave the default legacyexchangedn mapping **unchanged**. It will now work as designed.

Default mapping:

legacyexchangedn=~legacyexchangedn~/cn=Recipients/cn=~smtpname~.~smtphost~

Method #3 - Disable legdn detection

There are circumstances where detecting the destination legacyexchangedn root value is unnecessary. In this case, you can add an override to the connection config to disable this function.

- Go to the Custom tab in UnitySync
- Click the Raw Config button.
- Add a line with the value in the following format:

```
exch-legdn=disable
```

- Click Save.
- Click Save to save the connection.
- Run a Simulation to confirm results before running a Sync.

Custom Mapping: CSV Source

A CSV (or other delimited text file) can be used as a Source of information to update an LDAP directory. Use custom mapping as outlined below to sync the desired Source data.

Create a new connection. On the connection creation pop-up window, select Source Map Template type CSV. The default sync engine will also be CSV.

Configure the Source tab as described previously.

Each CSV Source file is unique and therefore default mappings are of limited use. The SMTP address and Name fields will be mapped successfully by default based on your input on the Source tab. The rest of the columns must use custom mapping. Use the actual CSV column names to customize your mappings. If your data file does not contain a header line, use the default headers of field0, field1, field2, etc.

Go to the Custom tab to create and configure your custom object map.

Once you create your custom object map, you'll need to edit the mappings using the following format:

```
DestinationAttribute=^CSVColumnName^
```

Change the field names on the right of the expression to match those of your CSV column headers. Alternatively, you can set up your CSV data file so that the column headers match those listed in the default map file.

IMPORTANT NOTE: When a column header contains a space character, the space is converted to a dash character for the purpose of attribute definition and mapping. If the Source header is Last Name or Last_Name, you must reference this column by specifying Last-Name as that is what the attribute name will be in the ldif.txt file. In fact, the allowed character set of column header values are limited to:

a-z A-Z 0-9 and - (dash)

In a Source CSV column header, any characters outside this set are converted to - (dash). This applies to field definitions on the Source tab as well as mappings in your custom object map or custom list map file.

The initial map file generated for you upon creation of your connection will include a few standard mappings which may need to be modified. However, if you fill in the Source tab correctly, you may leave the default mappings that utilize internal variables (denoted by tildes ~ as in ~mail~). The remainder of the mappings are commented out with a # character at the beginning of the line. To incorporate these mappings, remove the preceding # character and customize the column name as instructed above.

Custom Mapping: LDIF output

The Discovery phase of all connections outputs to an LDIF file. The Discovery phase does *not* utilize any mappings. All Source attributes are simply output to a raw LDIF file.

In order to utilize mapping/logic when producing ldif.txt, you can create a new connection, selecting Destination ENGINE type of "LDIF".

Fill in the source tab as usual. Fill in the Destination tab with a Placement DN corresponding with the root of *your Source discovery*. (In other words, you want the FQDN structure for each record in the export.txt to match the original ldif.txt.) To maintain levels, set levels=all. (Struct Name is not valid for this configuration. Do NOT fill it in.)

Customize your mappings as you would for a normal connection.

On Discovery, an LDIF.txt file will be generated as usual.

On SYNC, a new ldif file (named export.txt) will be generated. You should see the resulting export.txt shows the desired changes to mapped values.

Custom Mapping: ODBC

ODBC Source

An ODBC Data Source may be used as a Source of information to update any LDAP directory. This allows syncing with just about any database.

Create a new connection. On the connection creation pop-up window, select Source Map Template type ODBC and Source Sync Engine type ODBC.

See Importing from a Source Database via ODBC Data Source for details on how to create an ODBC Data Source for use with your UnitySync connection.

See ODBC Source Tab Configuration for details on how to configure your Source tab options.

Default mapping for an ODBC Source connection is limited to Name and Email address. This is because your Source columns are unique to your environment and therefore our default map file can not include the specific mappings required.

You must define the custom mapping to identify your Source columns and how they should map to the Destination attributes. To get started, create a custom object map file.

When your connection has an ODBC Source, your custom map file entries are in the following format:

```
DestinationAttribute=^Column-Name^
```

IMPORTANT NOTE: When a column header contains a space character, the space is converted to a dash character for the purpose of attribute definition and mapping. If the Source header is Last Name or Last_Name, you must reference this column by specifying Last-Name as that is what the attribute name will be in the ldif.txt file. In fact, the allowed character set of column header values are limited to:

a-z A-Z 0-9 and - (dash)

In a Source ODBC column header, any characters outside this set are converted to - (dash). This applies to field definitions on the Source tab as well as mappings in your custom object map or custom list map file.

The initial map file generated for you upon creation of your connection will include a few standard mappings which may need to be modified. However, if you fill in the Source tab correctly, you may leave the default mappings that utilize internal variables (denoted by tildes ~ as in ~mail~). The remainder of the mappings are commented out with a # character at the beginning of the line. To incorporate these mappings, remove the preceding # character and customize the column name as instructed above.

ODBC Destination

An ODBC Data Source may be used as a Destination. This allows syncing with just about any database type.

Create on new connection. On the connection creation pop-up window, select the Destination Map Template type ODBC as well as the Sync Engine type ODBC.

See ODBC Destination Tab Configuration for details on how to configure your Destination tab.

Attribute mapping for an ODBC Destination connection requires customization. The default mappings provided should be considered placeholders and must be modified to meet your attribute mapping needs. This is because all ODBC Destinations are unique. UnitySync has no way of knowing what columns appear on your Destination.

You must define the custom mappings to identify your Destination columns and how they should map from the Source attributes.

Once you create your custom object map, you'll need to edit the mappings using the following format:

```
ODBCColumnName=^SourceAttribute^
```

Change the field names on the left of the expression to match those of your ODBC column headers.

IMPORTANT NOTE: When a column header contains a space character, the space is converted to a dash character for the purpose of attribute definition and mapping. If the Source header is Last Name or Last_Name, you must reference this column by specifying Last-Name as that is what the attribute name will be in the `ldif.txt` file. In fact, the allowed character set of column header values are limited to:

a-z A-Z 0-9 and - (dash)

In a Destination ODBC column header, any characters outside this set are converted to - (dash). This applies to field definitions on the Source tab as well as mappings in your custom object map or custom list map file.

Appendix 3: Filters & Exclusions

Filters & Exclusions

Filters and Exclusions help you to limit the data that is synced to your Destination. Filters work on Discovery and thus, if a filter applies, the object will not appear in the Discovery log or ldif.txt file. Exclusions work on the Sync phase so you will see the information in the Discovery log and ldif.txt file, but it will not be applied at Sync time. If you have any questions as to whether to use either Filters or Exclusions on a connection, feel free to contact our Technical Support Team to discuss your goals and we'll be happy to recommend a solution.

Filters: Optional LDAP and o365 Source Query Filter

This information applies to an LDAP Directory Source as well as an o365 Source. (not ODBC, LDIF, CSV etc).

Custom Filters may be added to include and/or exclude objects based on certain values. Attribute filters can be placed on the object types listed below. If you are pulling multiple Source object types, it may be appropriate to specify the same filter in more than one filter parameter, or the filters may be different for each object type:

User: Filter entered here will be applied to Source objects of type User.

Contacts: Filter entered here will be applied to Source objects of type Contact.

Groups: Filter entered here will be applied to Source objects of type Group.

Public Folders: Filter entered here will be applied to Source Public Folders.

Syntax Overview

Understanding the following Operator Symbols will aid you in forming your LDAP query strings.

Operator	Means	Use when:
&	AND	All specified filters must be true for the statement to be true. For example, (&(filter)(filter)(filter))
\		OR
!	NOT	The specified statement must NOT be true for the statement to be true. Note that only one filter is affected by the NOT operator.

Keep in mind that Boolean expressions are evaluated in the following order:

1. innermost to outermost parenthetical expressions first
2. all expressions from left to right

Query Examples

Filter	Select when:
(manager=*)	The manager attribute is populated
(sn=Jones)	Entries with Surname of Jones
(sn=A*)	Entries when Surname begins with A
(mail=HC4)	HC4 is in the SMTP address
!(mail=*@acme.com))	Exclude any SMTP w/ this domain
!(st=Florida))	Exclude entries when state = Florida
(&(department=sales)(l=McLean))	Entries from sales AND in McLean

Actual attributes and objects vary by directory.

If you need assistance forming your query, please contact our Technical Support Team for assistance.

Filters Office 365 (O365)

To Discover an o365 source, you may use the standard LDAP formatted filters as described above. Here are a few additional tips specific to o365 filters.

To Discover o365 Users/contacts based on Group membership, refer to O365 Discovery - Query on Group Membership.

Exclusions

Exclusions are very important to the directory synchronization process. Exclusions are used to filter or limit the information you move between directories.

IMPORTANT NOTE Any time you are making changes to parameters that may affect the scope of the Source, like Exclusions, it is very important to disable Sync and enable Discovery and Simulation only. Run the connection with the Log File level set to 3-Detailed to confirm desired results before re-enabling Sync.

To add Exclude rules

1. Open the desired connection, select the Custom tab and click the Exclude button.
2. Add a single Exclude rule on each line.
3. Click Save.

Exclusion Rules

- Each line in the Exclude file is considered a rule, one rule per line. *The beginning of the rule defines the LDAP attribute to compare to (i.e. mail, dn, cn).
- A colon (:) should be used in most cases, as this is a basic string compare. An equal sign (=) may be used if you truly want to look for an **exact** value. Wild cards (*) and question marks (?) are not accepted.
- The exclusion using a colon simply checks to see if the string is embedded ANYWHERE in the attribute.
- All exclusion rules are case insensitive (case does NOT matter).

Uses for Exclusions

Character	Description
Attribute:Value to Exclude	This performs an exclude if the value appears anywhere in the attribute data
Attribute=Exact Value to Exclude	This performs an exclude if the exact value is the attribute data.
Mail:ExcludeThisDomain	This performs an exclude if the domain appears in the primary mail value.
Description: Do Not Sync	This performs and exclude if 'Do Not Sync' appears anywhere in the attribute value. You can do this with any available attribute.
Company=ABC Co.	This performs an exclude if the company value matches exactly
DN:ou=ContainerX	Excludes any instance of ou=ContainerX
DN:cn=John	Exclude a single, specific object
Smith,ou=Users,dc=domain,dc=com	

Exclusion Scenarios

Avoid looping when you are syncing bi-directionally:

When you are running a synchronization between two or more directories you will want to avoid returning entries back to their original Source. For example, if DirectoryA sends to DirectoryB, DirectoryB should not return the entries originating on DirectoryA. To do this you'll want to exclude the Sync Container via an exclusion rule.

For Active Directory and other LDAP directories

The rule is in this format:

```
DN:ou=SyncContainer
```

Fully qualified (FQDN) example:

```
DN:ou=SyncContainer,ou=Parent,dc=domain,dc=com
```

Exclude specific OU's

You may wish to exclude other Source containers from your sync for another reason. In this case, the format is exactly the same as described above. We recommend using the FQDN. For example, your Source has a container, Local Admins, which you want to exclude from your Sync:

```
DN:ou=Local Admins,ou=Users,dc=domain,dc=com
```

Exclude on null value

There are times you may want to exclude an object if a specific attribute is blank. Use this format:

```
sn=null
```

IMPORTANT NOTE: *The use of the = character is required, as is the text 'null.'* This exclude is only applied on the Sync phase. You will see Discovery still pulls object with the null SN, however Sync Excludes will prevent the object from being added to the Destination.

Special Excludes

By default, all Excludes are applied quietly; they do not produce an error at Sync time. The objects to be excluded are identified and simply ignored, and the Sync will end with no errors reported. Generally, this is the desired functionality.

In some cases, you may want instances of Excludes to cause the Sync process to log an error condition, so you are alerted to the fact a Special Exclude has taken place.

IMPORTANT NOTE *If Simulation is enabled, it will log instances of these Special Excludes but will **not** abort the Sync phase.*

To implement a Special Exclude, precede the exclude rule with an exclamation mark character, as shown:

```
!company:ABC
```

When a Sync processes this Special Exclude the Sync phase will throw an error, alerting the UnitySync Admin that any instance of company=ABC was detected and excluded during the Sync. The rest of the Sync will process normally assuming no other errors are reported.

Appendix 4: Automating your UnitySync Connections

To make synchronizing simpler, automate synchronization runs with a command line program (e.g. shell.exe) and any task scheduling program. Windows Task Scheduler is one of the most common scheduling programs and works well for this purpose.

Important Note: *UnitySync System Server Requirements indicate that minimum Hard Disk Space needed is 500Mb for a sync to run successfully. We further recommend that you ensure >10Gb Hard Disk Space is available depending on the number of connections you regularly run, the number of objects you are syncing, and your logging levels. If you choose to automate your syncs, we strongly recommend that you also set up Log File Management to ensure your logs don't accumulate beyond a reasonable amount of time. Without processes in place to prevent low disk space conditions, you could run the risk of mass deletions on one or more connections.*

Automating (Scheduling) your UnitySync connections

You may easily automate (schedule) your UnitySync connections.

Initial creation and run of connections is typically performed in the UnitySync User Interface (UI). Thereafter, your UnitySync connections may be run manually from the UI or via a command line program (shell.exe).

Using the command line shell.exe option, you may automate the your Sync runs via the use of any task scheduling program. The most commonly used is Windows Task Scheduler. Simply create a simple script using the syntax described below. Run the script manually first to confirm good results. Then set up a scheduled task to execute your Sync script as desired.

Note about log file cleanup

We **strongly recommend** that you also set up Log File Management to ensure your logs don't accumulate beyond a reasonable amount of time (or take up too much disk space).

Scripting Connections to Run

There are just a few things to keep in mind when scripting your connections to run:

- Follow the guidelines below to create a batch script.
- Run the script manually to ensure desirable results.
- Create a scheduled task to execute the script on your chosen schedule.

Script file requirements:

- Change to the local drive where UnitySync is installed
- Change to the local directory of the Synchronization executable
- Be sure to put each command on a separate line
- If you group your connections in Folders, you must specify the Folder name when calling the connection (i.e., ... \Shell Folder\YourConnection)**
- **Note: For versions prior to v4.6, omit --conn i.e. shell.exe "Paris-to-Master"**

Example Script #1 - One command per connection

Here is an example of a script to kick off UnitySync connections. Remember, one command line per connection:

```
C:
cd \UnitySync\bin
shell.exe --conn "Paris-to-Master"
shell.exe --conn "Denver-to-Master"
shell.exe --conn "NY-to-Master"
shell.exe --conn "Master-to-Paris"
shell.exe --conn "Master-to-Denver"
```

```
shell.exe --conn "Master-to-NY"
```

Here is an example of the same connections if they are in two folders (In/Out) instead of using naming conventions

```
C:
cd \UnitySync\bin
shell.exe --conn "IN\Paris-to-Master"
shell.exe --conn "IN\Denver-to-Master"
shell.exe --conn "IN\NY-to-Master"
shell.exe --conn "OUT\Master-to-Paris"
shell.exe --conn "OUT\Master-to-Denver"
shell.exe --conn "OUT\Master-to-NY"
```

Example Script #2 - Multiple connections per command

Here is an example of a batch script to kick off a subset of UnitySync connections with a single command. This is useful as it allows the creation of new connections to be inserted into the automation automatically, simply by use of proper naming conventions. No need to edit your script to insert the shell.exe command for each individual connection.

```
C:
cd \UnitySync\connections
for /D %%D in (in*.*) DO ..\bin\shell --conn "%%D"
for /D %%D in (out*.*) DO ..\bin\shell --conn "%%D"
```

In the above example, naming conventions are used to group connections. All connections that load the master (hub) are prefixed with "in". All connections that push from the master out to other directories, are prefixed with "out". Using this two line batch script, any connection created with a prefix of "in" or "out" are automatically included in the execution of the scheduled batch.

Below is an example of the same connections if they are in folders, instead of using naming conventions:

```
C:
cd \UnitySync\connections
for /D %%D in (IN\*.*) DO ..\bin\shell --conn "%%D"
for /D %%D in (OUT\*.*) DO ..\bin\shell --conn "%%D"
```

IMPORTANT NOTE: *If you want to test Advanced scripts from the command line, please note that you will need to substitute %D for %%D when running directly from the command line. Or, you may cd into the directory where the batch file exists and run that file from the command line for testing.*

Example Script #3 - Automating Forcemods

As noted in the knowledge base article detailing Forcemods functionality, this feature may also be automated.

To run from the command line:

```
shell.exe --forcemods --forcestruct --conn "Connection Name"
```

Use the same format for batch scripts as previously shown:

```
C:
cd \UnitySync\connections
shell.exe --forcemods --forcestruct --conn "Connection Name"
```

Please note:

- Forcemods allows Delete/Modify of all Destination person objects.
- Forcestruct allows recreation of deleted structure.

Remember, you **do not** want to add Forcemods to your batch script that runs daily. Determine how often you want to run Forcemods and create an additional script and task for it to run only occasionally, say weekly or monthly.

Scheduling Batch Execution

Using Scheduled Task (recommended)

- Create a Scheduled Task - Programs > Control Panel > Scheduled Task > Add a Scheduled Task.
- Select the script and configure run time and Run As properties. Be sure the Run As account has sufficient access/permissions to perform the operation.

IMPORTANT NOTE: Executing via Scheduled Task will pop up a CMD window. If you wish to run the Scheduled Task in the background, be sure to set the 'Run As' to an account other than the logged in account.

Using the AT command

Start the Scheduler Service - Start > Settings > Control Panel > Services. Or, configure Startup to make the service start at boot up.

Let's say that your batch file is called `...\UnitySync\runsync.bat` and you would like to have it run every day at 9:00 am. For syntax on the AT command type `at /` from the CMD prompt. Example:

```
at \\MyServer 9:00 /EVERY:SU,M,T,W,TH,F,S "C:\UnitySync\runsync.bat"
```

To view the scheduled events type `at`. Remember, a log file is generated with each run. Review these logs regularly to ensure your connection is running properly.

IMPORTANT NOTE: *UnitySync* is single thread so more than one occurrence of a program can run at a time. Take your bandwidth into consideration when attempting to run more than one sync.