



# AD\_Integrations 4me, ITRP - Active Directory Import

Creator: techwork data gmbh - Georg Leitner-Fidler  
Job title: Principal consultant

Version 1.02  
Date: 20/12/2017

# Contents

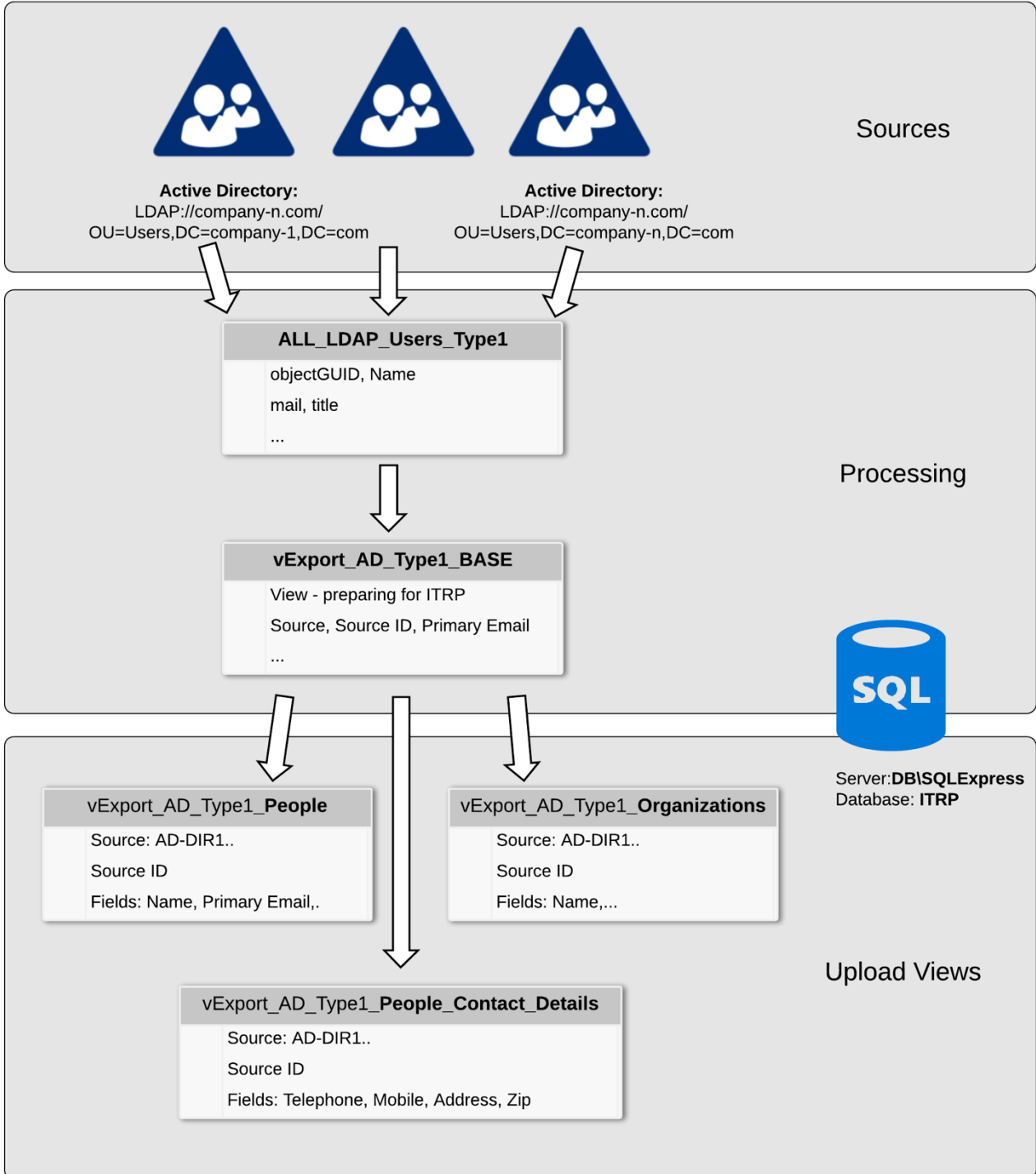
<b>REQUIREMENTS</b>	<b>3</b>
<b>OVERVIEW</b>	<b>4</b>
<b>FIELD MAPPING EXAMPLE</b>	<b>5</b>
<b>ACTIVE DIRECTORY (SOURCES)</b>	<b>5</b>
<b>SCRIPT FOLDER</b>	<b>6</b>
<b>SQL SETTINGS</b>	<b>7</b>
<b>ITRP SETTINGS</b>	<b>7</b>
<b>LDAP - ACTIVE DIRECTORY SETTINGS</b>	<b>8</b>
<b>SCHEDULING</b>	<b>9</b>
<b>INTERMEDIATE DB</b>	<b>9</b>

# Requirements

- Windows "Integration-Server": Host for PowerShell Scripts
  - MS SQL Management Studio
  - Access to the Database below 2)
  - Access to:
    - <https://api.4me.com/>\*
    - <https://api.4me.qa/>\*
    - <https://api.itrp.com/>\*
    - <https://api.itrp.qa/>\*
    - <https://itrp-eu.s3-eu-west-1.amazonaws.com/>\*
  - Access to all source Active Directories
- MS SQL or SQL-Express Database (2008R2 or later)
  - can be but not necessarily on the same machine
  - if not on the same machine, the machine above still requires MS SQL Management Studio
- Admin User
  - User on Windows "Integration Server" for Installation/Setup of Scripts
  - Access to Database 2) with "Database Owner" role
- Service User
  - "Password never expires"
  - User on Windows "Integration Server" for Scheduling of PowerShell Scripts
  - Access to Database 2) with "Database Owner" role

# Overview

## Active Directory to ITRP Integration



All configured Active Directories are downloaded and consolidated into ALL\_LDAP\_Users\_Type1.

Views provide the data to match the new target state.

# Field Mapping Example

ITRP	LDAP	Comment
<b>People</b>		
Source	AD- plus the ID from LDAP Configuration	e.g. 'AD-DIR1
Source ID	objectGUID	unique ID - stays the same if user is renamed
Primary Email	mail	primary email address from AD
Name	name	the full name from AD
Support ID	samAccountName	the short windows login name
Job Title	title	
Manager	manager	mapped to Primary Email of manager
Information	path hierarchy company department	combine some information from Active Directory into one field
<b>Contact Details:</b>		
telephone (work)	telephoneNumber	Telephone number (General Tab)
telephone (mobile)	mobile	Mobile (Telephones Tab)
Address (mailing)	streetAddress	Street (Address Tab)
City (mailing)	l	City (Address Tab)
State (mailing)	st	State/province (Address Tab)
Zip (mailing)	postalCode	Zip/Postal Code (Address Tab)
Country (mailing)	c	Country/region (Address Tab)

## Active Directory (Sources)

LDAP Configuration Section:

`$ldap_accounts` - array of LDAP access configurations

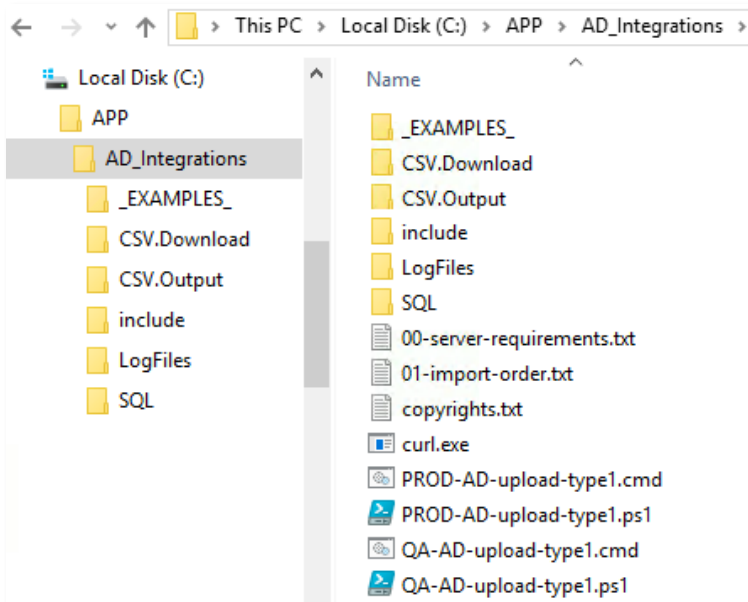
<b>id</b>	short name used as prefix in tables
<b>company</b>	company in ITRP for the ad data to be linked
<b>user</b>	user name to access over LDAP
<b>passwd</b>	password to access over LDAP
<b>integrated</b>	'true' or 'false', 'false' if above credentials should be used
<b>ssl</b>	'true' or 'false'
<b>path</b>	LDAP path e.g. LDAP://company.com/OU=All-Users,DC=company,DC=com
<b>filter</b>	LDAP filter: e.g. all accounts with an email address which are not disabled: ( & (objectclass=person) (mail=*) (!(userAccountControl:1.2.840.113556.1.4.803:=2)) )

# Script Folder

Host: ITRP01

User: ITRP01\NTRP\_User

Path: C:\APP\AD\_Integrations



The PowerShell scripts are hosted and run on a Windows machine dedicated for ITRP Integrations:

Files, Views and Functions containing 'QA' in their name are used for QA only.

Files, Views and Functions containing 'PROD' in their name are used for the Production Environment respectively.

SQL programming is always saved in files first, before executing and creating objects in the Intermediate database. SQL Views, Functions must not be applied without having them saved in files first.

Path	Remarks
Scripts (Main Script Folder)	<p>Custom PowerShell and batch scripts. Batch scripts (window .cmd files) are used for <b>scheduling</b> of PowerShell scripts</p> <ul style="list-style-type: none"> <li>• QA-AD-upload-type1.cmd (scheduled QA Integrations)</li> <li>• PROD-AD-upload-type1.cmd (scheduled Production Integrations)</li> </ul> <p>The following PowerShell script files contain sections with <b>configuration data</b> (Server, User,...) for LDAP, ITRP and SQL</p> <ul style="list-style-type: none"> <li>• QA-AD-upload-type1.ps1 Upload of people to ITRP QA</li> <li>• PROD-AD-upload-type1.ps1 Upload of people to ITRP Production</li> </ul>
Scripts\CSV.Download	<p>contains all intermediate download data from Active Directory and ITRP.</p> <p>useful for troubleshooting</p>

Scripts\CSV.Output	Based on 'Upload Views' data is stored here as CSV before being uploaded to ITRP via the ITRP Import API.  useful for troubleshooting
Scripts\include	Pre-build techwork PowerShell libraries for SQL, ITRP, LDAP prepared include files for upload of * people * people_contact_details * products * cis...  <i>(you should not need to make changes here)</i>
Scripts\LogFiles	Each PowerShell script from the main folder creates a log here. The log file name contains a timestamp of the script start time.
Scripts\SQL	Pre-build techwork SQL library functions and procedures  <i>(you should not need to make changes here)</i>

## SQL Settings

```
#####  
# SQL Settings #  
#####  
$ServerName = "Database\SQLExpress"  
$DatabaseName = "ITRP"  
$SQLUser = ""  
$SQLpwd = ""  
$SQLintegrated = "true"  
  
$deleteMode = "DROP"  
  
$importPrefix = "ITRP_PROD"  
$exportPrefix = "vExport_AD_Type1"
```

in above settings, the importPrefix must be different for Production and QA

## ITRP Settings

```
#####  
# ITRP Settings #  
#####  
$apiURL = "https://api.itrp.com/v1"  
  
# make sure you change the apiToken for each account!  
$useToken = "true"  
$apiToken = "tokenkentokenkentokenkentoken" # integration user  
  
$ITRPAccount = "widget"  
  
# Use these credentials for a Production-Environment  
$apiUsername = ""  
$apiPassword = ""  
  
$itrp_uploadMode = 'simple'  
$itrpSkipAccTables = "false"
```

in above settings, the apiToken, ITRPAccount and apiURL must be set for QA / Production

# LDAP - Active Directory Settings

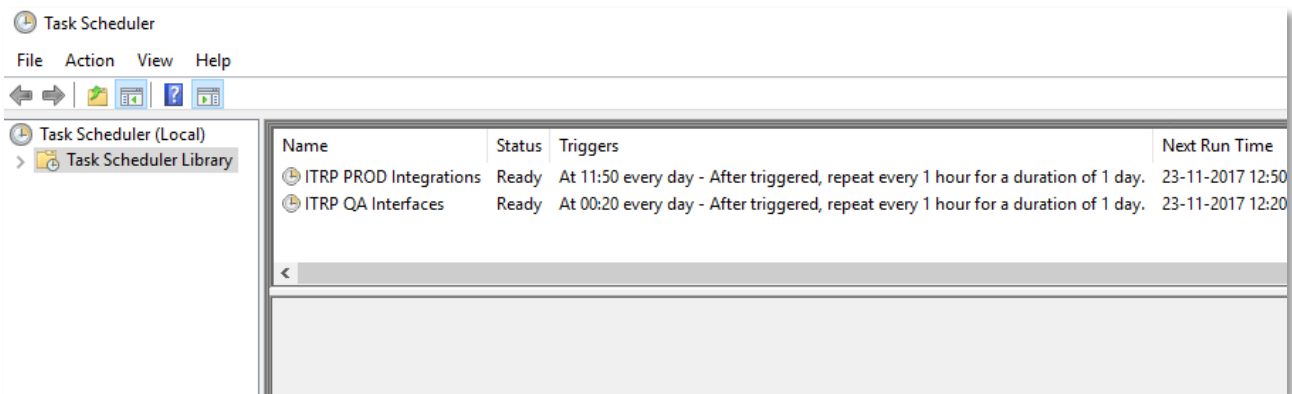
```
#####  
# LDAP Settings #  
#####  
  
$importLdapType = 'LDAP_Users_Type1';  
$aggregatePrefix = 'ALL';  
$ldap_uploadMode = "full_aggregate"  
$ldap_skipAccTables= 'false';  
  
$ldap_accounts =  
{  
  @(  
    @{"id="DIR1";company="TW1";user="user";passwd="passwd";integrated="true"  
    @{"id="DIR2";company="TW2";user="user";passwd="passwd";integrated="true"  
  )  
}
```

Each line in above settings represents one Active Directory Path. Each line can be temporarily deactivated by putting a # in front of the line.



# Scheduling

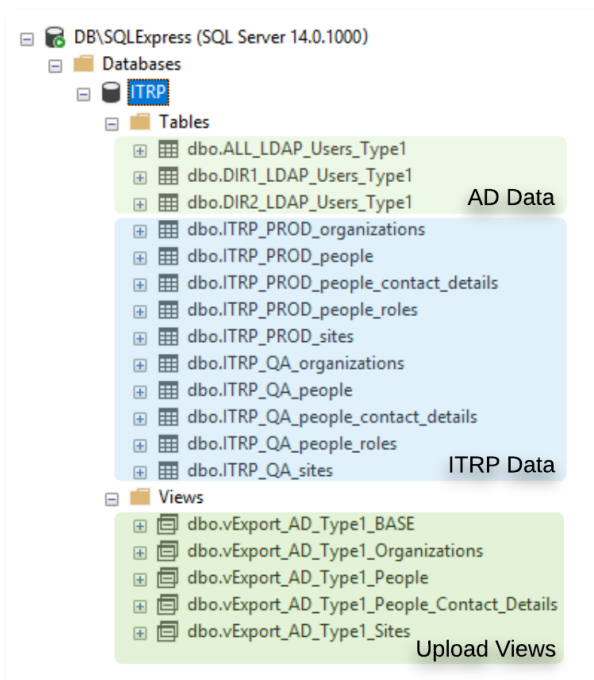
- **Production** (PROD-AD-upload-type1.cmd)  
when: e.g. every day at 08:00
- **QA** (QA-AD-upload-type1.cmd)  
when: e.g. every day at 07:30



# Intermediate DB

The Intermediate DB stores data downloaded from Active Directory and ITRP. Views and Functions provide the ITRP Upload with new data.

ITRP data must be downloaded before each upload so the upload include script can compare the current data in ITRP with the desired data from the 'Upload Views'.



## Example: People

- AD is downloaded to ALL\_LDAP\_Users\_Type1
- ITRP People are downloaded into ITRP\_QA\_people
- The new data is in vExport\_AD\_Type1\_people

include file for people upload does this:

- Upload of what has changed
- Disable what is no longer in vExport\_AD\_Type1\_people

Basically, for each module (people, people\_contact\_details, cis, products,) there is an include file for incremental data upload.