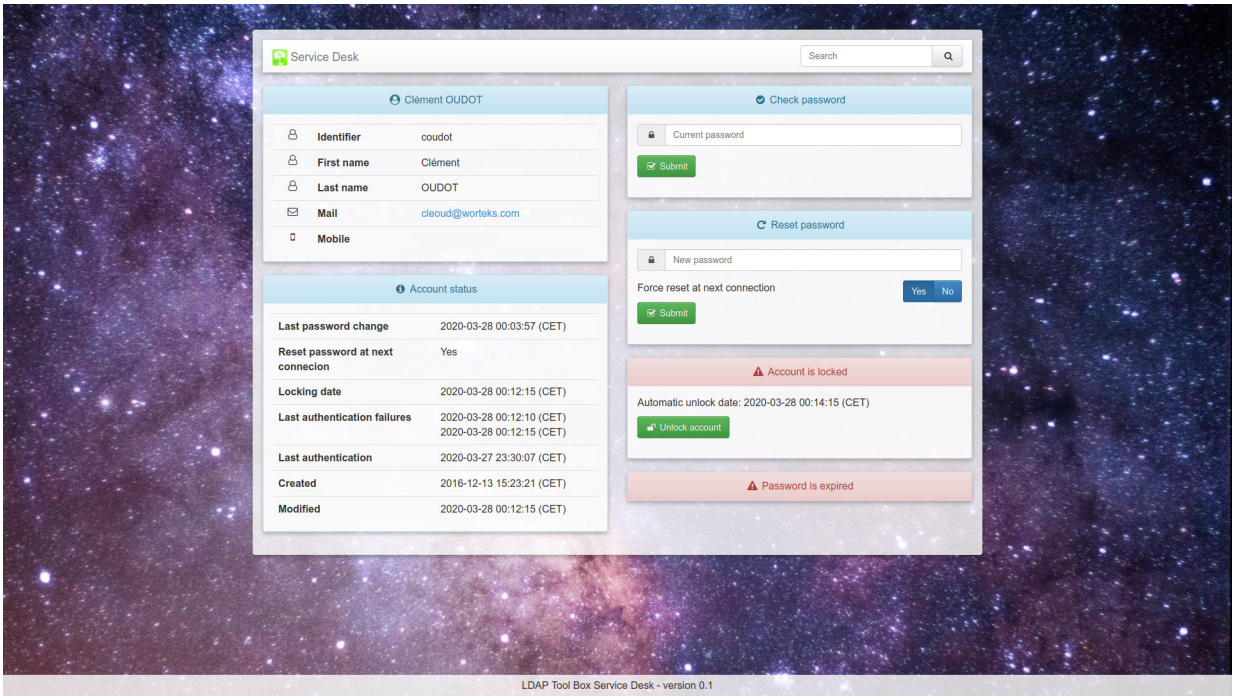
Service Desk Documentation

Clément OUDOT

Sep 09, 2020

Contents:



LDAP Tool Box Service Desk - version 0.1

CHAPTER 1

Presentation

LDAP Tool Box Service Desk is a web application for administrators and support teams. It allows to browse accounts in an LDAP directory, view and update their status.

Warning: There is no authentication requested by the application. You must set some before opening the application to your trusted users. Some examples are provided in the documentation.

1.1 Features

- Quick search for an account
- View main attributes
- View account and password status
- Test current password
- Reset password and force password change at next connection
- Lock and unlock account

CHAPTER 2

Installation

2.1 From tarball

Uncompress and unarchive the tarball:

```
$ tar -zxvf ltb-project-service-desk-*.tar.gz
```

Install files in `/usr/share/`:

```
# mv ltb-project-service-desk-* /usr/share/service-desk
```

You need to install these prerequisites:

- Apache or another web server
- php
- php-ldap
- smarty (version 3)

2.2 Debian / Ubuntu

Configure the repository:

```
# vi /etc/apt/sources.list.d/ltb-project.list
```

```
deb [arch=amd64] https://ltb-project.org/debian/stable stable main
```

Import repository key:

```
# wget -O - https://ltb-project.org/wiki/lib/RPM-GPG-KEY-LTB-project | sudo apt-key add -
```

Then update:

```
# apt update
```

You are now ready to install:

```
# apt install service-desk
```

2.3 CentOS / RedHat

> **Warning:** You may need to install first the package php-Smarty which is not in official repositories.

Configure the yum repository:

```
# vi /etc/yum.repos.d/ltb-project.repo
```

```
[ltb-project-noarch]
name=LTB project packages (noarch)
baseurl=https://ltb-project.org/rpm/$releasever/noarch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-LTB-project
```

Then update:

```
# yum update
```

Import repository key:

```
# rpm --import https://ltb-project.org/wiki/lib/RPM-GPG-KEY-LTB-project
```

You are now ready to install:

```
# yum install service-desk
```

CHAPTER 3

Apache configuration

3.1 Virtual host

Here is a sample virtual host configuration:

```
<VirtualHost *:80>
   ServerName sd.example.com

   DocumentRoot /usr/share/service-desk/htdocs
   DirectoryIndex index.php

   <Directory /usr/share/service-desk/htdocs>
       AllowOverride None
       Require all granted
   </Directory>

   LogLevel warn
   ErrorLog /var/log/apache2/sd_error.log
   CustomLog /var/log/apache2/sd_access.log combined
</VirtualHost>
```

Tip: The application can also be published in a directory inside the default host

Warning: You must protect the access to the application, else everyone will be able to reset any user password!

3.2 LDAP authentication and authorization

You can use Apache mod_authnz_ldap. This module checks user credentials against the LDAP directory and can restrict access to users or groups.

```
<Directory /usr/share/service-desk/htdocs>
    AllowOverride None
    AuthType basic
    AuthName "LTB Service Desk"
    AuthBasicProvider ldap
    AuthLDAPURL ldap://ldap.example.com/dc=example,dc=com?uid
    Require ldap-group cn=support,ou=groups,dc=example,dc=com
</Directory>
```

3.3 External authentication

You can use any authentication source and authentication protocols, like CAS, SAML or OpenID Connect. Configuring these solutions is out of scope of the current documentation.

CHAPTER 4

General parameters

4.1 Configuration files

The default configuration file is `conf/config.inc.php`, it contains all default values. To edit configuration, you should create `conf/config.inc.local.php` and override needed parameters:

```
<?php
// Override config.inc.php parameters below

?>
```

Warning: Do not copy `config.inc.php` into `config.inc.local.php`, as the first one includes the second. You would then create an infinite loop and crash your application.

4.2 Language

Tip: Lang is selected from browser configuration. If no matching language is found, the default language is used.

Set default language in `$lang`:

```
$lang = "en";
```

Tip: You can override messages by creating lang files in `conf/`, for example `conf/en.inc.php`.

4.3 Dates

You can adapt how dates are displayed with specifiers (see strftime reference):

```
$date_specifiers = "%Y-%m-%d %H:%M:%S (%Z)";
```

4.4 Graphics

4.4.1 Logo

You change the default logo with your own. Set the path to your logo in `$logo`:

```
$logo = "images/ltb-logo.png";
```

4.4.2 Background

You change the background image with your own. Set the path to image in `$background_image`:

```
$background_image = "images/unsplash-space.jpeg";
```

4.4.3 Custom CSS

To easily customize CSS, you can use a separate CSS file:

```
$custom_css = "css/custom.css";
```

4.4.4 Footer

You can hide the footer bar:

```
$display_footer = false;
```

4.5 Debug

You can turn on debug mode with `$debug`:

```
$debug = true;
```

4.6 Smarty

You need to define where Smarty is installed:

```
define("SMARTY", "/usr/share/php/smarty3/Smarty.class.php");
```

For Smarty to be able to successfully 'compile' pages from the templates, the directory "template_c" must be writable by the web server.

CHAPTER 5

LDAP parameters

5.1 Server address

Use an LDAP URI to configure the location of your LDAP server in `$ldap_url`:

```
$ldap_url = "ldap://localhost:389";
```

You can set several URI, so that next server will be tried if the previous is down:

```
$ldap_url = "ldap://server1 ldap://server2";
```

To use SSL, set ldaps in the URI:

```
$ldap_url = "ldaps://localhost";
```

If RACF Manager is used, you may need to specify a different LDAP URL for the LDAP Server SDBM backend using `$sdbm_ldap_url`:

```
$sdbm_ldap_url = "ldaps://zvmsys.example.com";
```

To use StartTLS, set `true` in `$ldap_starttls`:

```
$ldap_starttls = true;
```

Tip: LDAP certificate management in PHP relies on LDAP system libraries. Under Linux, you can configure `/etc/ldap.conf` (or `/etc/ldap/ldap.conf` on Debian/Ubuntu, or `C:\OpenLDAP\sysconf\ldap.conf` for Windows). Provide the certificate from the certificate authority that issued your LDAP server's certificate.

5.2 Credentials

Configure DN and password in `$ldap_bindn` and `$ldap_bindpw`:

```
$ldap_binddn = "cn=manager,dc=example,dc=com";
$ldap_bindpw = "secret";
```

Tip: You can use the LDAP admin account or any service account. The account needs to read users, password policy entries and write `userPassword` and `pwdReset` attributes in user entries. Note that using the LDAP admin account will bypass any password policy like minimal size or password history when reseting the password.

5.3 LDAP Base

You can set global base in `$ldap_base`:

```
$ldap_base = "dc=example,dc=com";
```

In a RACF environment (with `$racf_mode = true`), the LDAP tree pointed to by `$ldap_base` should use a "standard" RFC 2307-style schema. This would be, say, your z/VM LDAP LDBM with Native Authenticaation, or OpenLDAP with slapo-rwm to redirect binds to SDBM. Your RACF SDBM will be at a different base, and some operations need this base to work directly on SDBM. Set this using `$sdbm_base`:

```
$sdbm_base = "o=ZVMSSI1";
```

Setting `$sdbm_base` also enables the RACF Manager.

5.4 User search parameters

You can set base of the search in `$ldap_user_base`:

```
$ldap_user_base = "ou=users,".$ldap_base;
```

The filter can be set in `$ldap_user_filter`:

```
$ldap_user_filter = "(objectClass=inetOrgPerson)";
```

5.5 Size limit

It is advised to set a search limit on client side if no limit is set by the server:

```
$ldap_size_limit = 100;
```

5.6 Default password policy

Set `$ldap_default_ppolicy` value if a default policy is configured in your LDAP directory.

```
$ldap_default_ppolicy = "cn=default,ou=ppolicy,dc=example,dc=com";
```

Tip: Password policy is first searched in `pwdPolicySubentry` attribute of user entry, then fallback to default policy.

CHAPTER 6

Attributes

Attributes are defined in `$attributes_map`, where each item is an array with these keys:

- `attribute`: name of LDAP attribute, in lower case
- `faclass`: name of Font Awesome icon class
- `type`: type of attribute (text, mailto, tel or date)

This is used to configure how attribute is displayed.

Available types:

- `text`: simple text
- `mailto`: mailto link
- `tel`: tel link
- `boolean`: true or false
- `date`: full date
- `list`: value from a list
- `bytes`: bytes converted in KB/MB/GB/TB

Tip: See LDAP Tool Box White Pages documentation to get more information.

CHAPTER 7

Search parameters

7.1 Search attributes

Configure attributes on which the search is done:

```
$search_attributes = array('uid', 'cn', 'mail');
```

By default, search is done with substring match. This can be changed to use exact match:

```
$search_use_substring_match = false;
```

7.2 Results display

Configure items shown when displaying results:

```
search_result_items = array('identifier', 'mail', 'mobile');
```

7.3 Datatables

Define pagination values in dropdown:

```
$datatables_page_length_choices = array(10, 25, 50, 100, -1); // -1 means All
```

Set default pagination for results (can also be used to force the length without `$datatables_page_length_choices`):

```
$datatables_page_length_default = 10;
```

Enable or disable autoPrint feature:

```
$datatables_auto_print = true;
```

CHAPTER 8

Display parameters

8.1 User panel

Configure which items are displayed:

```
$display_items = array('identifier', 'firstname', 'lastname', 'title',
↪'businesscategory', 'employeenumber', 'employeetype', 'mail', 'phone', 'mobile',
↪'fax', 'postaladdress', 'street', 'postalcode', 'l', 'state', 'organizationalunit',
↪'organization');
```

Set which item is use as title:

```
$display_title = "fullname";
```

Choose to show undefined values:

```
$display_show_undefined = false;
```

8.2 Account information panel

Configure which items are displayed:

```
$display_password_items = array('pwdchangedtime', 'pwdreset', 'pwdaccountlockedtime',
↪'pwdfailuretime','pwdpolicysubentry', 'authtimestamp', 'created', 'modified');
```

CHAPTER 9

Check password

This feature allows to enter a password and check authentication.

> **Warning:** the authentification can fail even if the password is correct. This is currently the case if account is locked or password is expired. We are developing checks for these states, so over time the correct status will be presented. For example, an expired RACF password now reports with a warning “expired” message.

To enable this feature:

```
$use_checkpassword = true;
```

CHAPTER 10

Reset password

This feature allows to reset a password and set the reset at next connection flag.

To enable this feature:

```
$use_resetpassword = true;
```

When changing the password, you can force the user to reset it at next connection. To configure the default value presented in the form:

```
$resetpassword_reset_default = true;
```

CHAPTER 11

Lock account

This feature allows to lock the account permanently. The button is only displayed if the account is not locked.

To enable this feature:

```
$use_lockaccount = true;
```

CHAPTER 12

Unlock account

This feature allows to unlock the account. It is only displayed if the account is already locked.

To enable this feature:

```
$use_unlockaccount = true;
```

CHAPTER 13

Posthook

Posthook feature allows to run a script after the password modification.

The script is called with two parameters: login and new password.

13.1 Parameters

Define posthook script (and enable the feature):

```
$posthook = "/usr/share/service-desk/posthook.sh";
```

Define which attribute will be used as login:

```
$posthook_login = "uid";
```

Display posthook error:

```
$display_posthook_error = true;
```

Encode passwords sent to posthook script as base64:

```
$posthook_password_encodebase64 = true;
```

Tip: This will prevent alteration of the passwords if set to true. To read the actual password in the posthook script, use a base64_decode function/tool.