
Service Desk Documentation

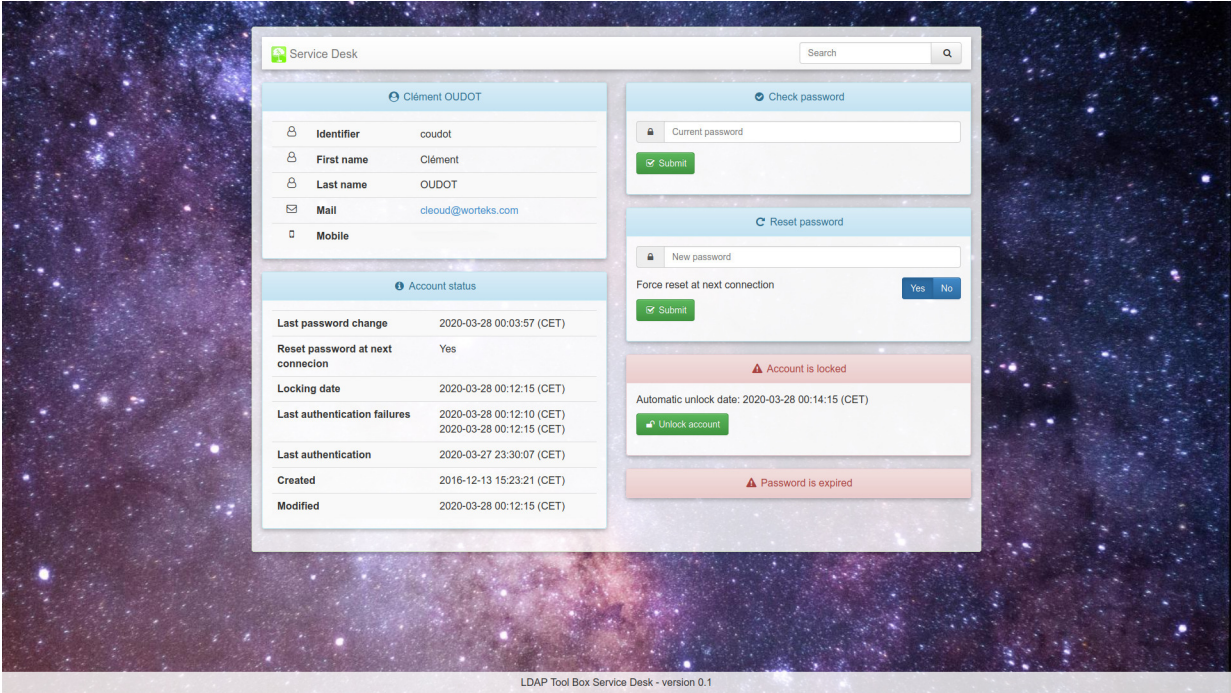
Clément OUDOT

May 17, 2023

Contents:

1	Presentation	3
1.1	Features	3
2	Installation	5
2.1	From tarball	5
2.2	Debian / Ubuntu	5
2.3	CentOS / RedHat	6
2.4	Docker	6
3	Apache configuration	7
3.1	Virtual host	7
3.2	LDAP authentication and authorization	7
3.3	External authentication	8
4	General parameters	9
4.1	Configuration files	9
4.2	Multi tenancy	9
4.3	Language	10
4.4	Dates	10
4.5	Graphics	10
4.6	Debug	11
4.7	Smarty	11
4.8	Notify administrator by mail	11
5	LDAP parameters	13
5.1	Server address	13
5.2	Credentials	13
5.3	LDAP Base	14
5.4	User search parameters	14
5.5	Size limit	14
5.6	Default password policy	14
6	Attributes	15
7	Search parameters	17
7.1	Search attributes	17
7.2	Results display	17

7.3	Datatables	17
8	Display parameters	19
8.1	User panel	19
8.2	Account information panel	19
9	Check password	21
10	Reset password	23
10.1	Notify user by mail	23
10.2	Notify administrator by mail	23
11	Lock account	25
12	Unlock account	27
13	Hook	29
13.1	Parameters	29
14	Dashboards	31
14.1	Locked accounts	31
14.2	Soon expired passwords	31
14.3	Expired passwords	31
14.4	Idle accounts	32
15	Mail	33
15.1	LDAP Attributes	33
15.2	Sender name	33
15.3	Change password notification	33
15.4	PHPMailer	34
16	Audit	35
16.1	Audit log file	35
16.2	Admin name	36



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
- Dashboards:
 - Accounts locked
 - Accounts with a password that will soon expire
 - Accounts with an expired password
 - Accounts idle (never connected or not connected since a number of days)

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/documentation/_static/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/documentation/_static/RPM-GPG-KEY-LTB-project
```

You are now ready to install:

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

2.4 Docker

Prepare a local configuration file for Service Desk, for example `/home/test/servicedesk.conf.php`.

Start container, mounting that configuration file:

```
# docker run -p 80:80 -v /home/test/servicedesk.conf.php:/var/www/conf/config.inc.local.php
```

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.

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 Multi tenancy

You can load a specific configuration file by passing a HTTP header. This feature is disabled by default. To enable it:

```
$header_name_extra_config = "SSP-Extra-Config";
```

Then if you send the header `SSP-Extra-Config: domain1`, the file `conf/config.inc.domain1.php` will be loaded.

Using Apache, we may set such header using the following:

```
<VirtualHost *:80>
    ServerName ssp.domain1.com
    RequestHeader setIfEmpty SSP-Extra-Config domain1
    [...]
</VirtualHost>
```

Using Nginx, we could use instead:

```
server {
    [...]
    location ~ /\.php {
        fastcgi_param HTTP_SSP_EXTRA_CONFIG domain1;
        [...]
    }
}
```

4.3 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`.

4.4 Dates

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

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

4.5 Graphics

4.5.1 Logo

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

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

4.5.2 Background

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

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

4.5.3 Custom CSS

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

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

4.5.4 Footer

You can hide the footer bar:

```
$display_footer = false;
```

4.6 Debug

You can turn on debug mode with \$debug:

```
$debug = true;
```

Tip: Debug messages will be printed in server logs.

This is also possible to enable Smarty debug, for web interface issues:

```
$smarty_debug = true;
```

Tip: Debug messages will appear on web interface.

4.7 Smarty

You need to define where Smarty is installed:

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

4.8 Notify administrator by mail

It is possible to provide mail of administrator to service-desk using a HTTP header.

\$header_name_notify_admin_by_mail is name of header that will be provided to cgi script as HTTP_\$header_name_notify_admin_by_mail to set administrator mail from webserver.

```
$header_name_notify_admin_by_mail = "SSP-Admin-Mail";
```

Using Apache, we may set such header using the following:

```
<VirtualHost *:80>
    ServerName ssp.domain1.com
    RequestHeader setIfEmpty SSP-Admin-Mail admin@example.com
    [...]
</VirtualHost>
```

Using Nginx, nginx take normalized cgi param naming, ie uppercase and - replaced to _. we could use instead:

```
server {  
    [...]   
    location ~ /\.php {  
        fastcgi_param HTTP_SSP_ADMIN_MAIL admin@example.com;  
        [...]   
    }  
}
```

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";
```

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 `$ldap_binddn` and `$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 resetting the password.

5.3 LDAP Base

You can set global base in `$ldap_base`:

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

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.

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

- `attribute`: name of LDAP attribute, in lower case
- `faiclass`: 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`: LDAP date converted to full date
- `list`: value from a list
- `bytes`: bytes converted in KB/MB/GB/TB
- `timestamp`: timestamp converted to full date
- `dn_link`: convert DN into link to account display page
- `address`: convert address string to multi-lines

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

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;
```

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');
```

You can also display the password expiration date in this panel by enabling this option:

```
$display_password_expiration_date = true;
```


CHAPTER 9

Check password

This feature allows to enter a password and check authentication.

Warning: the authentication can fail even if the password is correct. This is the case if account is locked or password is expired.

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;
```

If you do not want to let the choice to reset at next connection, you can hide this button:

```
$use_resetpassword_resetchoice = false;
```

In this case, the value set in `$resetpassword_reset_default` will be applied.

10.1 Notify user by mail

You can notify the user by mail when the password is reset:

```
$notify_on_change = true;
```

To do so you need to configure mail (see [Mail](#)).

10.2 Notify administrator by mail

`$notify_admin_by_mail_list` is a php array list of mails of admins to set directly in service-desk configuration.

`$header_name_notify_admin_by_mail` is name of header that will be provided to cgi script as `HTTP_$header_name_notify_admin_by_mail` to set administrator mail from webserver. This setting allows

webserver to select an administrator mail related to current http session. Its value should be a valid mail according to rfc822

If those are set any administrator mail will receive a notification that user password was changed.

Remark this is not controled by \$notify_on_change, which is only for user.

```
$notify_admin_by_mail_list=array( 'a@example.com', 'b@example.com' 'c@example.com');  
$header_name_notify_admin_by_mail=SD_ADMIN_MAIL;
```

To do so you need to configure mail (see [Mail](#)).

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;
```


Hook feature allows to run a script before or after the password modification.

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

13.1 Parameters

Define prehook or posthook script (and enable the feature):

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

Define which attribute will be used as login:

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

You can choose to display an error if the script return code is greater than 0:

```
$display_prehook_error = true;  
$display_posthook_error = true;
```

The displayed message will be the first line of the script output.

Another option can be enabled to encode the password in base64 before sending it to the script, which can avoid an execution issue if the password contains special characters:

```
$prehook_password_encodebase64 = false;  
$posthook_password_encodebase64 = false;
```

By default with prehook script, the password will not be changed in LDAP directory if the script fails. You can change this behavior to ignore script error. This could be useful to run prehook script and display a warning if it fails, but still try to update password in the directory.

```
$ignore_prehook_error = true;
```

14.1 Locked accounts

This page will list all currently locked accounts.

To enable it:

```
$use_searchlocked = true;
```

It is possible to unlock an account directly from this page. This requires to enable the feature *Unlock account*.

14.2 Soon expired passwords

This page will list all accounts with a password that will expire in the next days.

To enable it:

```
$use_searchwillexpire = true;
```

You can also configure the number of days before expiration:

```
$willexpiredays = 14;
```

14.3 Expired passwords

This page will list all accounts with an expired password.

To enable it:

```
$use_searchexpired = true;
```

14.4 Idle accounts

This page will list all accounts never connected, or not connected since a number of days.

Tip: This requires the `authTimestamp` attribute which is provided by the `lastbind` overlay.

To enable it:

```
$use_searchidle = true;
```

You can also configure the number of idle days:

```
$idledays = 60;
```

15.1 LDAP Attributes

Set the LDAP attributes where user email may be stored:

```
$mail_attributes = array( "mail", "gosaMailAlternateAddress", "proxyAddresses" );
```

Tip: Only the first value of this attribute will be used to get the mail address.

You can also define which LDAP attribute contains the user name (will be used in mail body content):

```
$mail_username_attribute = "cn";
```

15.2 Sender name

You can change the default From header and add a signature:

```
$mail_from = "admin@example.com";  
$mail_from_name = "Self Service Password administrator";  
$mail_signature = "";
```

15.3 Change password notification

Use this option to send a confirmation mail to the user, just after a successful password change:

```
$notify_on_change = true;
```

This option is false by default.

15.4 PHPMailer

You have to set all parameters for PHPMailer if you use mail notification.

```
$mail_sendmailpath = '/usr/sbin/sendmail';
$mail_protocol = 'smtp';
$mail_smtp_debug = 0;
$mail_debug_format = 'html';
$mail_smtp_host = 'localhost';
$mail_smtp_auth = false;
$mail_smtp_user = '';
$mail_smtp_pass = '';
$mail_smtp_port = 25;
$mail_smtp_timeout = 30;
$mail_smtp_keepalive = false;
$mail_smtp_secure = 'tls';
$mail_smtp_autotls = true;
$mail_smtp_options = array();
$mail_contenttype = 'text/plain';
$mail_wordwrap = 0;
$mail_charset = 'utf-8';
$mail_priority = 3;
```

Tip: See <https://github.com/PHPMailer/PHPMailer> for more information

CHAPTER 16

Audit

You can enable audit to log all actions done through Service Desk.

The items provided in the audit log are:

- Date
- IP of connected admin
- DN of account being updated
- Who has done the action (see Admin name below)
- Action
- Result of the action

Example:

```
{
  "date": "Wed, 17 May 2023 11:12:59",
  "ip": "127.0.0.1",
  "user_dn": "uid=donald,ou=users,dc=example,dc=com",
  "done_by": "Mickey",
  "action": "lockaccount",
  "result": "accountlocked"
}
```

16.1 Audit log file

Set the file where actions are logged:

```
$audit_log_file = "/var/log/service-desk/audit.log";
```

Tip: The file must be writable by the PHP or WebServer process

16.2 Admin name

The admin name must be set into an HTTP header.

```
$header_name_audit_admin = "Auth-User";
```

Using Nginx, we could use instead:

```
$header_name_audit_admin = "AUTH_USER";
```

In nginx.conf:

```
server {
    [...]
    location ~ /\.php {
        fastcgi_param HTTP_AUTH_USER $http_auth_user;
        [...]
    }
    [...]
}
```

Warning: Using Nginx, headers with underscores in their names are discarded by default. In order for these headers to be considered valid, we need to add `underscores_in_headers on` to `nginx.conf`.

Tip: If no header defined or if header is empty, actions will be logged as “anonymous”
