



- [Downloads](#)
- [Documentation](#)
- [Get Involved](#)
- [Help](#)

[PHP 8.0.0 Release Candidate 1 available for testing](#)

## [Getting Started](#)

[Introduction](#)

[A simple tutorial](#)

## [Language Reference](#)

[Basic syntax](#)

[Types](#)

[Variables](#)

[Constants](#)

[Expressions](#)

[Operators](#)

[Control Structures](#)

[Functions](#)

[Classes and Objects](#)

[Namespaces](#)

[Errors](#)

[Exceptions](#)

[Generators](#)

[References Explained](#)

[Predefined Variables](#)

[Predefined Exceptions](#)

[Predefined Interfaces and Classes](#)

[Context options and parameters](#)

[Supported Protocols and Wrappers](#)

## [Security](#)

[Introduction](#)

[General considerations](#)

[Installed as CGI binary](#)

[Installed as an Apache module](#)

[Session Security](#)

[Filesystem Security](#)

[Database Security](#)

[Error Reporting](#)

[Using Register Globals](#)

[User Submitted Data](#)

[Magic Quotes](#)

[Hiding PHP](#)

[Keeping Current](#)

## [Features](#)

[HTTP authentication with PHP](#)

[Cookies](#)

[Sessions](#)

[Dealing with XForms](#)

[Handling file uploads](#)

[Using remote files](#)

[Connection handling](#)

[Persistent Database Connections](#)  
[Safe Mode](#)  
[Command line usage](#)  
[Garbage Collection](#)  
[DTrace Dynamic Tracing](#)

## [Function Reference](#)

[Affecting PHP's Behaviour](#)  
[Audio Formats Manipulation](#)  
[Authentication Services](#)  
[Command Line Specific Extensions](#)  
[Compression and Archive Extensions](#)  
[Cryptography Extensions](#)  
[Database Extensions](#)  
[Date and Time Related Extensions](#)  
[File System Related Extensions](#)  
[Human Language and Character Encoding Support](#)  
[Image Processing and Generation](#)  
[Mail Related Extensions](#)  
[Mathematical Extensions](#)  
[Non-Text MIME Output](#)  
[Process Control Extensions](#)  
[Other Basic Extensions](#)  
[Other Services](#)  
[Search Engine Extensions](#)  
[Server Specific Extensions](#)  
[Session Extensions](#)  
[Text Processing](#)  
[Variable and Type Related Extensions](#)  
[Web Services](#)  
[Windows Only Extensions](#)  
[XML Manipulation](#)  
[GUI Extensions](#)

## Keyboard Shortcuts

?	This help
j	Next menu item
k	Previous menu item
g p	Previous man page
g n	Next man page
G	Scroll to bottom
g g	Scroll to top
g h	Goto homepage
g s	Goto search (current page)
/	Focus search box

[ldap\\_bind\\_ext »](#)

[« ldap\\_add\\_ext](#)

- [Manual do PHP](#)
- [Referência das Funções](#)
- [Outros Serviços](#)
- [LDAP](#)
- [LDAP Funções](#)

Change language: Brazilian Portuguese ▼[Edit Report a Bug](#)

## ldap\_add

(PHP 4, PHP 5, PHP 7)

ldap\_add — Add entries to LDAP directory

### Descrição ¶

**ldap\_add** ( resource \$link\_identifier , string \$dn , array \$entry [, array \$serverctrls = array() ] ) : bool

Add entries in the LDAP directory.

### Parâmetros ¶

link\_identifier

An LDAP link identifier, returned by [ldap\\_connect\(\)](#).

dn

The distinguished name of an LDAP entity.

entry

An array that specifies the information about the entry. The values in the entries are indexed by individual attributes. In case of multiple values for an attribute, they are indexed using integers starting with 0.

```
<?php
$entry["attribute1"] = "value";
$entry["attribute2"][0] = "value1";
$entry["attribute2"][1] = "value2";
?>
```

serverctrls

Array of [LDAP Controls](#) to send with the request.

### Valor Retornado ¶

Retorna **TRUE** em caso de sucesso ou **FALSE** em caso de falha.

### Changelog ¶

Versão	Descrição
--------	-----------

Versão	Descrição
7.3	Support for serverctrls added

## Exemplos ¶

### Exemplo #1 Complete example with authenticated bind

```
<?php
$ds = ldap_connect("localhost"); // assuming the LDAP server is on this host

if ($ds) {
    // bind with appropriate dn to give update access
    $r = ldap_bind($ds, "cn=root, o=My Company, c=US", "secret");

    // prepare data
    $info["cn"] = "John Jones";
    $info["sn"] = "Jones";
    $info["objectclass"] = "person";

    // add data to directory
    $r = ldap_add($ds, "cn=John Jones, o=My Company, c=US", $info);

    ldap_close($ds);
} else {
    echo "Unable to connect to LDAP server";
}
?>
```

## Notas ¶

**Nota:** Esta função é binary-safe.

## Veja Também ¶

- [ldap\\_add\\_ext\(\)](#) - Add entries to LDAP directory
- [ldap\\_delete\(\)](#) - Delete an entry from a directory

 [add a note](#)

## User Contributed Notes 23 notes

[up](#)  
[down](#)

10

[Axel D. \(FRANCE\) ¶](#)

17 years ago

Try this script if you don't know how to add an user in the AD Win2K.

To have more informations about the attributes, open the adsiedit console in the Support Tools for Win2K.

```
$adduserAD["cn"][0] =
$adduserAD["instancetype"][0] =
$adduserAD["samaccountname"][0] =
$adduserAD["objectclass"][0] = "top";
$adduserAD["objectclass"][1] = "person";
$adduserAD["objectclass"][2] = "organizationalPerson";
$adduserAD["objectclass"][3] = "user";
```

```

$adduserAD["displayname"][0] =
$adduserAD["name"][0] =
$adduserAD["givenname"][0] =
$adduserAD["sn"][0] =
$adduserAD["company"][0] =
$adduserAD["department"][0] =
$adduserAD["title"][0] =
$adduserAD["description"][0] =
$adduserAD["mail"][0] =
$adduserAD["initials"][0] =
$adduserAD["samaccountname"][0] =
$adduserAD["userprincipalname"][0] =
$adduserAD["profilepath"][0] =
$adduserAD["manager"][0] = ***Use DistinguishedName***

if (!$ldap = ldap_connect("localhost")) {
    die ("Could not connect to LDAP server");
}
if (!$res = @ldap_bind($ldap, "user@pc.com", $password)) {
    die ("Could not bind to the LDAP account");
}
if (!ldap_add($ldap, "CN=New User,OU=OU Users,DC=pc,DC=com", $adduserAD)){
    echo "There is a problem to create the account
    echo "Please contact your administrator !";
    exit;
}
ldap_unbind($ldap);

```

[up](#)  
[down](#)

5

[damien at groovey dot com ¶](#)

**13 years ago**

Here is how to add a user with a hashed MD5 password to OpenLDAP. I used this technique to migrate Drupal accounts into OpenLDAP for a single-sign-on solution.

The trick to it is to tell OpenLDAP the hash type (e.g. {MD5}) before the password, and also to base64 encode the BINARY hashed result. You cannot just base64 encode what is returned by PHP's md5() or sha() hash functions, because they return a hexadecimal text string. First you must use pack("H\*", \$hash\_result) to make that a binary string, THEN you can base64 encode it.

Here is complete code for connecting and adding a user with a hashed password. You don't have to use {MD5}, you could pick a different hash if that is what you have. The output from one of these hashed passwords will look like this: {md5}bdwD04RS9xMDGVi1n/H36Q==

Finally some caveats: This technique will not work if you hashed the password using a salt value (but Drupal does not). This technique will also certainly not work with active directory, where passwords can definitely only be set over SSL connections and hashing probably works differently.

---- snip ----

```

$ds = ldap_connect($serverAddress);
if ($ds) {
    ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3); // otherwise PHP defaults to ldap v2 and you will get a
    Syntax Error!
    $r = ldap_bind($ds, $managerDN, $managerPassword);
    $ldaprecord['cn'] = $newuser_username;
    $ldaprecord['givenName'] = $newuser_firstname;
    $ldaprecord['sn'] = $newuser_surname;

```

```
// put user in objectClass inetOrgPerson so we can set the mail and phone number attributes
$ldaprecord['objectclass'][0] = "person";
$ldaprecord['objectclass'][1] = "organizationalPerson";
$ldaprecord['objectclass'][2] = "inetOrgPerson";
$ldaprecord['mail'] = $newuser_email_address;
$ldaprecord['telephoneNumber'] = $newuser_phone_number;
// and now the tricky part, base64 encode the binary hash result:
$ldaprecord['userPassword'] = '{MD5}' . base64_encode(pack('H*', $newuser_md5hashed_password));
// If you have the plain text password instead, you could use:
// $ldaprecord['userPassword'] = '{MD5}' . base64_encode(pack('H*', md5($newuser_plaintext_password)));
$r = ldap_add($ds, $base_user_dn, $ldaprecord);
} else { die "cannot connect to LDAP server at $serverAddress."; }
```

[up](#)  
[down](#)

4

[stian](#)

**12 years ago**

This solution works for us.

In the form the CN and pwdtxt are randomly generated from strict rules.

This script creates 50-60 users i AD pr.day! and never even had a glitch!

```
<?php
## From form
$CN = $_POST['CN'];
$givenName = $_POST['givenName'];
$SN = $_POST['SN'];
$mail = $_POST['mail'];
$Phone = $_POST['Phone'];
$pwdtxt = $_POST['pwdtxt'];

$AD_server = "localhost:390"; // Local Stunnel -> http://www.stunnel.org/
$AD_Auth_User = "administrator@student.somewhere.com"; //Administrative user
$AD_Auth_PWD = "duppidupdup"; //The password

$dn = 'CN='.$CN.',OU=Brukere,DC=student,DC=somewhere,DC=com';

## Create Unicode password
$newPassword = "\". $pwdtxt . "\"";
$len = strlen($newPassword);
$newPassw = "";

for($i=0;$i<$len;$i++) {
    $newPassw .= "{$newPassword{$i}}\000";
}

## CONNNECT TO AD
$ds = ldap_connect($AD_server);
if ($ds) {
    ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3); // IMPORTANT
    $r = ldap_bind($ds, $AD_Auth_User, $AD_Auth_PWD); //BIND

    $ldaprecord['cn'] = $CN;
    $ldaprecord['givenName'] = $givenName;
    $ldaprecord['sn'] = $SN;
    $ldaprecord['objectclass'][0] = "top";
    $ldaprecord['objectclass'][1] = "person";
    $ldaprecord['objectclass'][1] = "organizationalPerson";
    $ldaprecord['objectclass'][2] = "user";
```

```

$ldaprecord['mail'] = $mail;
$ldaprecord['telephoneNumber'] = $Phone;
$ldaprecord['unicodepwd'] = $newPassw;
$ldaprecord['sAMAccountName'] = $CN;
$ldaprecord['UserAccountControl'] = "512";
//This is to prevent the user from beeing disabled. -->
http://support.microsoft.com/default.aspx?scid=kb;en-us;305144

```

```

$r = ldap_add($ds, $dn, $ldaprecord);

} else {
    echo "cannot connect to LDAP server at $AD_server.";
}

?>

```

This is code example creates a user i AD.  
 We use this on an internal web page to create  
 temporary users that can access the wireless network.  
 We have a .pl script that deletes the users after 24H.

[up](#)  
[down](#)

2

[ondrej dot duchon at t-systems dot cz ¶](#)  
**16 years ago**

jharnett at artschool dot com:

For active user in AD u must change "useraccountcontrol" to 512, 512 = enabled, 514 = disabled

[up](#)  
[down](#)

4

[phil at networkalliance dot com ¶](#)  
**12 years ago**

I created a simple function that can be called to create global distribution groups in Active Directory:

```

<?php
function ldap_createGroup($object_name, $dn, $members, $ldap_conn)
{
    $addgroup_ad['cn']="$object_name";
    $addgroup_ad['objectClass'][0] = "top";
    $addgroup_ad['objectClass'][1] ="group";
    $addgroup_ad['groupType']="2";
    $addgroup_ad['member']=$members;
    $addgroup_ad['sAMAccountName'] =$object_name;

    ldap_add($ldap_conn,$dn,$addgroup_ad);

    if(ldap_error($ldap_conn) == "Success")
        return true;
    else
        return false;
}

?>

```

You can call this function using the follow code:

```

<?php
$ldap_conn = ldap_bind();
$object_name="Test Group";

```

```
$dn="CN=".$object_name.",OU=PathToAddGroupTo,OU=All Users,DC=YOURDOMAIN,DC=COM";
$members[] ="CN=User1,OU=PathToAddGroupTo,OU=All Users,DC=YOURDOMAIN,DC=COM";
$members[] ="CN=User2,OU=PathToAddGroupTo,OU=All Users,DC=YOURDOMAIN,DC=COM";

ldap_createGroup($object_name, $dn, $members, $ldap_conn);
?>
```

The other function I created is ldap\_bind(), and this can be used to bind to an LDAP server:

```
<?php
function ldap_bind()
{
    $ldap_addr = '192.168.1.1'; // Change this to the IP address of the LDAP server
    $ldap_conn = ldap_connect($ldap_addr) or die("Couldn't connect!");
    ldap_set_option($ldap_conn, LDAP_OPT_PROTOCOL_VERSION, 3);
    $ldap_rdn = "domain_name\\user_account";
    $ldap_pass = "user_password";

    // Authenticate the user against the domain controller
    $flag_ldap = ldap_bind($ldap_conn,$ldap_rdn,$ldap_pass);
    return $ldap_conn;
}
?>
```

[up](#)  
[down](#)

3

[sergioshev ¶](#)

**12 years ago**

once i'am having problemes to add attributes with boolean syntax (1.3.6.1.4.1.1466.115.121.1.7)

```
$['boolean_attr']=true; //give me one warning, ldap_add(): Add: Invalid syntax
```

solved this by setting the value on this:

```
$['boolean_attr']='TRUE';
```

hope this can helps.

[up](#)  
[down](#)

1

[John Van Atta ¶](#)

**17 years ago**

In response to jharnett's question about accounts disabled by default from ldap\_add, we have found a solution.

The attribute userAccountControl contains a value that includes whether the account is disabled or enabled. The default for us is 546; when we changed that to 544 the account became enabled. Changing whatever value is in userAccountControl by 2 seems to enable or disable the account.

The following code worked for us to create a new user with an enabled account:

```
$adduserAD["userAccountControl"] = "544";
```

We just added this element to the above example's array.

[up](#)  
[down](#)

2

[theiderich AT laweekly dot com ¶](#)

**15 years ago**



When adding/editing attributes for a user, keep in mind that the 'memberof' attribute is a special case. The memberOf attribute is not an accessible attribute of the user schema. To add someone to a group, you have to add the user in the group, and not the group in the user. You can do this by accessing the group attribute 'member':

```
<?php
```

```
$group_name = "CN=MyGroup,OU=Groups,DC=example,DC=com";
$group_info['member'] = $dn; // User's DN is added to group's 'member' array
ldap_mod_add($connect,$group_name,$group_info);
```

```
?>
```

[up](#)

[down](#)

1

[hp at syntomax dot com ¶](#)

**15 years ago**

Another fun thing: ldap\_add() doesn't like arrays with empty members: so array (

```
    [cn] = "name"
```

```
    [key] = ""
```

```
    [anotherkey] = "value"
```

```
)
```

will yield a syntax error!

solve this with a simple peice of code:

```
foreach ($originalobject as $key => $value){
    if ($value != ""){
        $object[$key] = $value;
    }
}
```

where \$originalobject is the unchecked array and \$object is the one without empty members.

[up](#)

[down](#)

1

[ondrej dot duchon at t-systems dot cz ¶](#)

**16 years ago**

IF you need use national characters (iso 8859-2,8 etc.) it's good way to use ldap\_set\_option. It was hard job to find where is a bug ;-)). I hope that helps somebody.

```
ldap_set_option($ldap, LDAP_OPT_PROTOCOL_VERSION, 3);
```

[up](#)

[down](#)

0

[Baumkuchen.TH ¶](#)

**1 year ago**

Create Group in Active Directory

```
<?php
```

```
$ds = ldap_connect("IP-server/localhost");
```

```
$base_dn = "CN=Group name,OU=Organization Unit,DC=Domain-name,DC=com";//distinguishedName of group
```

```
if ($ds) {
```

```
// bind with appropriate dn to give update access
```

```
ldap_bind($ds, "CN=Administrator,OU=Organization Unit,DC=Domain-name,DC=com", "some-password");
```

```
//Add members in group
```

```
$member_array = array();
$member_array[0] = "CN=Administrator,OU=Organization Unit,DC=Domain-name,DC=com";
$member_array[1] = "CN=User,OU=Organization Unit,DC=Domain-name,DC=com";
```

```
$entry["cn"] = "GroupTest";
$entry["samaccountname"] = "GroupTest";
$entry["objectClass"] = "Group";
$entry["description"] = "Group Test!!";
$entry["member"] = $member_array;
$entry["groupType"] = "2";//GroupType="2" is Distribution / GroupType="1" is Security
```

```
ldap_add($ds,$base_dn,$entry);
```

```
ldap_close($ds);
} else {
echo "Unable to connect to LDAP server";
}
?>
```

[up](#)

[down](#)

1

[akohlsmith at mixdown dot org](#)

**21 years ago**

ldap\_add() will only honour the \$entry["attribute"][x]="value" \*if there are multiple values for the attribute\*. If there is only one attribute value, it \*MUST\* be entered as \$entry["attribute"]="value" or ldap\_add() sets the value for the attribute to be "Array" instead of what you put into \$entry["attribute"][0].

Here is a little routine I wrote up to do this automatically. when you're parsing the input, just use multi\_add():

```
<?php
function multi_add($attribute, $value)
{
global $entry;                // the LDAP entry you're gonna add

if(isset($entry[$attribute]))
    if(is_array($entry[$attribute]))
        $entry[$attribute][count($entry[$attribute])] = $value;
    else
    {
        $tmp = $entry[$attribute];
        unset($entry[$attribute]);
        $entry[$attribute][0] = $tmp;
        $entry[$attribute][1] = $value;
    }
else
    $entry[$attribute] = $value;
}
?>
```

multi\_add() checks to see if there is already a value for the attribute. if not, it adds it as \$entry[\$attribute]=\$value. If there is already a value for the attribute, it converts the attribute to an array and adds the multiple values correctly.

How to use it:

```
<?php
switch($form_data_name)
{
case 'phone': multi_add("telephoneNumber", $form_data_value); break;
case 'fax': multi_add("facsimileTelephoneNumber", $form_data_value); break;
case 'email': multi_add("mail", $form_data_value); break;
...
}
```

```
}
?>
```

In the system I designed the form has pull downs with names ctype1, ctype2, ctype3, etc. and the values are "fax, mail, phone...". The actual contact data (phone number, fax, email, etc) is contact1, contact2, contact3, etc. The user pulls down what the contact type is (phone, email) and then enters the data (number, address, etc.)

I use variable variables to fill the entry and skip blanks. Makes for a very clean form entry system. email me if you're interested in it, as I think I'm outgrowing the size of note allowed here. :-)

[up](#)  
[down](#)

0

[paul90brown at gmail dot com ¶](#)

**7 years ago**

I kept getting "Object Class Violation" when I tried adding posixAccount and shadowAccount as an objectclass. It turned out that these object classes had a lot of required fields that I was not adding. You may need to export a working user (if you have phpLDAPadmin) and see exactly what fields they have, then try to copy it exactly in the script. It also doesn't hurt if you make everything an Array the first time around, you can fix those fields later.

[up](#)  
[down](#)

0

[chad dot smith at 50marketing dot com ¶](#)

**15 years ago**

I took spam2004 at turniton dot dk example and made it a bit better. Maybe my setup was a bit different but either way here is how I added a group in Microsoft Windows Server 2003.

```
<?php
// Connect using ldap_connect
// Bind using ldap_bind
// Set LDAP_OPT_PROTOCOL_VERSION to 3
$member_array = array();
$member_array[0] = "cn=user1,cn=Users,dc=yourdomain,dc=com";
$member_array[1] = "cn=administrator,cn=Users,dc=yourdomain,dc=com";

$addgroup_ad["cn"] = "testgroup";
$addgroup_ad["samaccountname"] = "testgroup";
$addgroup_ad["objectClass"] = "Group";
$addgroup_ad["description"] = "Yep just a test.";
$addgroup_ad["member"] = $member_array;
$base_dn = "cn=testgroup,cn=Users,DC=yourdomain,DC=com";
ldap_add($ldap_conn,$base_dn,$addgroup_ad);
// This is it.
?>
```

Take care and good luck,  
Chad R. Smith

[up](#)  
[down](#)

0

[amcnabb ¶](#)

**15 years ago**

Be careful with types. PHP switches automatically between strings and numbers, but LDAP doesn't, and PHP will send whatever is most convenient for PHP, not LDAP, unless you specify a type.

If you inadvertently send a number as a string, you will get an error: "ldap\_add(): Add: Invalid syntax in [filename] on line LINENUM."

Observe this example which makes an array to send to LDAP to create a POSIX group. Note that \$new\_groupid, which is technically a string, must be typecast with (int).

```
$new_ldap_group['cn'] = $groupname;
$new_ldap_group['objectclass'][0] = 'posixgroup';
$new_ldap_group['objectclass'][1] = 'top';
$new_ldap_group['gidnumber'] = (int) $new_groupid;
```

[up](#)[down](#)

0

[spam2004 at turniton dot dk ¶](#)**16 years ago**

To add a group in Windows AD..

```
$object_name="testgroup2";
$members[]="CN=THU,ou=Users,dc=addomain,dc=domain,dc=dk";
$members[]="CN=testgroup2,ou=Groups,dc=addomain,dc=domain,dc=dk";
$addgroup_ad['cn']="$object_name";
$addgroup_ad['objectClass'][0] = "top";
$addgroup_ad['objectClass'][1] ="group";
$addgroup_ad['description']=$object_description;
$addgroup_ad['member']=$members;
$addgroup_ad["sAMAccountName"] =$object_name;
```

// notice param 2 (dn) will probably be different

```
$dn="cn=".$object_name.",ou=Groups,dc=addomain,dc=domain,dc=dk";
ldap_add($ldapc,$dn,$addgroup_ad);
```

[up](#)[down](#)

0

[jharnett at artschool dot com ¶](#)**17 years ago**

For some wacky reason, when the new account is added using ldap\_add(), the account is set to "inactive".

And from what I can see there is no modifiable attribute to "re-enable" that user. I suppose by default, if the specific flags are not set the default values are used. Anyone that has a fix for this, please post, I'm pulling out what little hair I have left.

[up](#)[down](#)

0

[del at babel dot com dot au ¶](#)**18 years ago**

If you need to add an attribute that is binary encoded (eg: userCertificate), then you need to add the ";binary" specification at the end of the field name.

eg:

```
$info["userCertificate;binary"] = $myBinaryCert;
$ldap_add ...
```

Del

[up](#)[down](#)

0

[titus dot stahl at experts4 dot com ¶](#)**19 years ago**

Note that you cannot use base64 encoding, you have to use utf-8 encoding for special chars instead.

[up](#)[down](#)

-1

[micattack+phpnet at gmail dot com ¶](#)**15 years ago**

When getting the dreaded invalid syntax, it helps turning on debugging in ldap. Looking at /var/log/ldap, gets you things like

```
May  2 13:51:21 tux slapd[12985]: conn=4934 op=1 RESULT tag=105 err=21 text=phpgwtz: value #0 invalid per syntax
May  2 13:52:02 tux slapd[12697]: No objectClass for entry (uid=1, ou=adressen, dc=...
```

[up](#)  
[down](#)

-1

[Andrew \(a.whyte at cqu.edu.au\)](#)

**17 years ago**

In reference to the questions about Account Enabling, you can use the table found at Microsoft's Support site to help with these attributes.

You are correct that '2' is the Account Disabled flag, but there are others, which allow you to detect/set password force expiry and the like.

Hope this URL is usefull for that:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;305144>

Cheers.

[up](#)  
[down](#)

-5

[Igor2i](#)

**7 years ago**

Решил добавить специально для русскоязычных пользователей.

У меня сервер на Gentoo с поддержкой кириллицы в кодировке UTF-8.

И при вводе через форму кириллицы выдавал ошибку синтаксиса.

Решение проблемы:

1) код требуется перекодировать в UTF-8 без BOM

2) и в начале кода вставить строку

```
<?php
```

```
header('Content-Type: text/html; charset=utf-8');
```

```
?>
```

[up](#)  
[down](#)

-2

[Willie](#)

**13 years ago**

In response to:

Note that you cannot use base64 encoding, you have to use utf-8 encoding for special chars instead.

--

I have found that if I have a special character (or a newline) that ldap\_add and ldap\_modify will automatically do the base64 encoding for you. For example:

```
<?php
```

```
// assuming binding is done, etc.
```

```
$entry['postalAddress'] = "123 East 456 West\nSuite A103";
```

```
ldap_modify($ds, $dn, $entry);
```

```
?>
```

The function or server will take the newline and convert it into base64 automatically (same goes for other special characters).

You may be able to verify by using a command-line ldapsearch

```
ldapsearch -b "dc=example,dc=com" -x "(cn=Example Person)" postalAddress
```

You'll see that the result comes up as

```
dn: cn=Example Person,dc=example,dc=com
```

```
postalAddress:: MTIzIEVhc3QgNDU2IFdlc3QKU3VpdGUgQTEwMw==
```

See the double colons after postal address? That's how LDAP states it's base64 encoded in this case.

[+ add a note](#)

- [LDAP Funções](#)

- [ldap\\_8859\\_to\\_t61](#)
- [ldap\\_add\\_ext](#)
- [ldap\\_add](#)
- [ldap\\_bind\\_ext](#)
- [ldap\\_bind](#)
- [ldap\\_close](#)
- [ldap\\_compare](#)
- [ldap\\_connect](#)
- [ldap\\_control\\_paged\\_result\\_response](#)
- [ldap\\_control\\_paged\\_result](#)
- [ldap\\_count\\_entries](#)
- [ldap\\_delete\\_ext](#)
- [ldap\\_delete](#)
- [ldap\\_dn2ufn](#)
- [ldap\\_err2str](#)
- [ldap\\_errno](#)
- [ldap\\_error](#)
- [ldap\\_escape](#)
- [ldap\\_exop\\_passwd](#)
- [ldap\\_exop\\_refresh](#)
- [ldap\\_exop\\_whoami](#)
- [ldap\\_exop](#)
- [ldap\\_explode\\_dn](#)
- [ldap\\_first\\_attribute](#)
- [ldap\\_first\\_entry](#)
- [ldap\\_first\\_reference](#)
- [ldap\\_free\\_result](#)
- [ldap\\_get\\_attributes](#)
- [ldap\\_get\\_dn](#)
- [ldap\\_get\\_entries](#)
- [ldap\\_get\\_option](#)
- [ldap\\_get\\_values\\_len](#)
- [ldap\\_get\\_values](#)
- [ldap\\_list](#)
- [ldap\\_mod\\_add\\_ext](#)
- [ldap\\_mod\\_add](#)
- [ldap\\_mod\\_del\\_ext](#)
- [ldap\\_mod\\_del](#)
- [ldap\\_mod\\_replace\\_ext](#)
- [ldap\\_mod\\_replace](#)
- [ldap\\_modify\\_batch](#)
- [ldap\\_modify](#)
- [ldap\\_next\\_attribute](#)
- [ldap\\_next\\_entry](#)
- [ldap\\_next\\_reference](#)
- [ldap\\_parse\\_exop](#)
- [ldap\\_parse\\_reference](#)
- [ldap\\_parse\\_result](#)
- [ldap\\_read](#)

- [ldap\\_rename\\_ext](#)
- [ldap\\_rename](#)
- [ldap\\_sasl\\_bind](#)
- [ldap\\_search](#)
- [ldap\\_set\\_option](#)
- [ldap\\_set\\_rebind\\_proc](#)
- [ldap\\_sort](#)
- [ldap\\_start\\_tls](#)
- [ldap\\_t61\\_to\\_8859](#)
- [ldap\\_unbind](#)
- [Copyright © 2001-2020 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)