

Model HowTo

1. Create a new PHP file and name it the same as your LDAP objectclass.
2. Create a new PHP class within these file named as your LDAP objectclass prefixed by "Model_" and extending LDM
The objectclass must begin with a Capital

```
class Model_Objectclass extends LDM
{
...
}
```

3. Add the protected attribute \$_attributes as an associative array containing all LDAP attributes initialized with null.
Group the attributes by internal, required and optional with a leading comment.
Sort the attributes alphabetical.

```
...
/*
 * List of attributes
 */
protected $_attributes = array(
    //Internal
    'createTimestamp' => null,
    ...

    //Required
    'cn' => array(),
    ...

    //Optional
    'businessCategory' => array(),
    ...
);
...
```

4. Add the protected attribute \$_referers as array containing all LDAP attributes which holds a DN to another LDAP object.
Sort the attributes alphabetical.

```
...
/*
 * List of attributes holding a dn to an other object
 */
protected $_referers = array(
    'member',
    ...
);
...
```

5. Add the protected attribute \$_protected as array containing all LDAP attributes which are read only.
Group the attributes by internal, required and optional with a leading comment.
Sort the attributes alphabetical.

```
...
/*
 * protected attributes are read only
 */
protected $_protected = array(
    //Internal
    'createTimestamp',
    ...
);
...
```

```
);  
...
```

6. Add the protected attribute `$_multivalues` as array containing all LDAP attributes which can hold one or more values. Group the attributes by internal, required and optional with a leading comment. Sort the attributes alphabetical.

```
...  
/*  
 * List of multivalue attributes  
 */  
protected $_multivalues = array(  
    //Required  
    'cn',  
    ...  
  
    //Optional  
    'businessCategory',  
    ...  
);  
...
```

7. Add the protected attribute `$_object_class` containing the LDAP objectclass as string.

```
...  
/*  
 * The object class must be set on each model. It corresponds with the internal  
 * LDAP attribute 'structuralObjectClass'.  
 */  
protected $_object_class = 'objectclass';  
...
```

8. Add the protected attribute `$_primary_key` containing the RDN of the LDAP object as string.

```
...  
/*  
 * Primary key can be used when creating objects with factory class  
 * Example: $group = Dobject::factory('group', 'some value for the primary key');  
 * Note: The primary key must be an unique value  
 */  
protected $_primary_key = 'cn';  
...
```

9. Create an empty class extending the just created one with a shorter or more speakable name. This step is optional. Is is only used to make life easier while programming.

```
class Model_Shortname extends Model_Objectclass {}
```