Products.LDAPUserFolder Documentation

Release 3.2.dev0

Jens Vagelpohl

Nov 21, 2019

Contents

`Products.LDAPUserFolder` provides a user folder for the Zope web application server that uses LDAP to store and retrieve login data.

CHAPTER 1

API documentation

These interfaces describe the official programming API for `Products.LDAPUserFolder`.

Note: On ReadTheDocs this page will not work correctly.

1.1 `Products.LDAPUserFolder.interfaces.ILDAPUser`

1.2 `Products.LDAPUserFolder.interfaces.ILDAPUserFolder`

CHAPTER 2

Narrative documentation

Narrative documentation explaining how to use `Products.LDAPUserFolder`.

2.1 Installation

You will need Python version 2.7 to run `Products.LDAPUserFolder`.

`Products.LDAPUserFolder` requires the `pyldap` library. Make sure your system has LDAP development files and libraries installed so `pyldap` can build without failure.

If you use `zc.buildout` you can add `Products.LDAPUserFolder` to the necessary `eggs` section to have it pulled in automatically.

2.2 Frequently asked questions

2.2.1 General

Why use LDAP to store user records?

LDAP as a source of Zope user records is an excellent choice in many cases, like...

- You already have an existing LDAP setup that might store company employee data and you do not want to duplicate any data into a Zope user folder
- You want to make the same user database available to other applications like mail, address book clients, operating system authenticators (PAM-LDAP) or other network services that allow authentication against LDAP
- You have several Zope installations that need to share user records or a ZEO setup
- You want to be able to store more than just user name and password in your Zope user folder
- You want to manipulate user data outside of Zope

... the list continues.

What should my directory tree or schema look like?

Your LDAP server should contain records that can be used as user records. Any object types like person, organizationalPerson, or inetOrgPerson and any derivatives thereof should work. Records of type posixAccount should work correctly as well. The LDAPUserFolder expects your user records to have at least the following attributes, most of which are required for the abovementioned object classes, anyway:

- an attribute to hold the user ID (like cn, uid, etc)
- userPassword (the password field)
- objectClass
- whatever attribute you choose as the username attribute
- typcial person-related attributes like sn (last name), givenName (first name), uid or mail (email address) will make working with the LDAPUserFolder nicer

Zope users have certain roles associated with them, these roles determine what permissions the user have. For the LDAPUserFolder, role information can be expressed through membership in group records in LDAP.

Group records can be of any object type that accepts multiple attributes of type "uniqueMember" or "member" and that has a "cn" attribute. One such type is "groupOfUniqueNames". The cn describes the group / role name while the member attributes point back to all those user records that are part of this group. Only those group-style records that use full DNs for its members are supported, which excludes classes like posixGroup.

It is outside of the scope of this documentation to describe the different object classes and attributes in detail, please see LDAP documentation for a better treatment.

Help, I locked myself out of my Zope site!

Since a user folder is one of these items that can lock users out of the site if they break I suggest testing the settings in some inconspicuous location before replacing a site's main acl_users folder with a LDAPUserFolder. As a last resort you will always be able to log in and make changes as the superuser (or in newer Zope releases called "emergency user") who can delete and create user folders.

2.2.2 Microsoft Active Directory

In general, with ActiveDirectory *Your Mileage May Vary*. Neither do I have any Windows-based environment, nor any Windows-version with a running ActiveDirectory installation. I have fixed ActiveDirectory-related issues in the past, though, relying on feedback from users.

Are nested groups supported?

Nested groups as used by AD are not supported at this time.

Why does AD crash my Zope authentication?

It's not clear if this is still an issue, but some ActiveDirectory versions formatted LDAP query results in a way that was incompatible with the LDAPUserFolder product. As a workaround, instead of running LDAP queries through the default ports (389 or 636), ActiveDirectory offers the so-called "Global Catalog" on port 3268. Query responses from the "Global Catalog" were more correctly formatted. See https://www.mail-archive.com/activedir@mail.activedir.org/msg03887.html for details.

2.2.3 Netscape directory products

Why does the LDAPUserFolder not show all my LDAP groups?

Netscape Directory at some points allowed the creation of empty group records, meaning group records with no member attributes. Those records will not show up in the LDAPUserFolder. Only group records with at least one member attribute are considered.

2.3 Development

2.3.1 Getting the source code

The source code is maintained on GitHub. To check out the trunk:

```
$ git clone https://github.com/dataflake/Products.LDAPUserFolder.git
```

You can also browse the code online at https://github.com/dataflake/Products.LDAPUserFolder

2.3.2 Bug tracker

For bug reports, suggestions or questions please use the GitHub issue tracker at https://github.com/dataflake/Products.LDAPUserFolder/issues.

2.3.3 Running the tests using `zc.buildout`

`Products.LDAPUserFolder` ships with its own `buildout.cfg` file and `bootstrap.py` for setting up a development buildout:

```
$ python bootstrap.py
...
Generated script '.../bin/buildout'
$ bin/buildout
...
```

Once you have a buildout, the tests can be run as follows:

```
$ bin/test
Running tests at level 1
Running zope.testrunner.layer.UnitTests tests:
  Set up zope.testrunner.layer.UnitTests in 0.000 seconds.
  Running:
..............................................................
  Ran 62 tests with 0 failures and 0 errors in 0.043 seconds.
Tearing down left over layers:
  Tear down zope.testrunner.layer.UnitTests in 0.000 seconds.
```

2.3.4 Building the documentation using `zc.buildout`

The `Products.LDAPUserFolder` buildout installs the Sphinx scripts required to build the documentation, including testing its code snippets:

```
$ cd docs
$ make html
...
build succeeded.

Build finished. The HTML pages are in _build/html.
```

2.3.5 Making a release

These instructions assume that you have a development sandbox set up using `zc.buildout` as the scripts used here are generated by the buildout.

```
$ bin/buildout -o
$ python setup.py sdist bdist_wheel upload --sign
```

The `bin/buildout` step will make sure the correct package information is used.

2.4 Change log

2.4.1 3.2 (unreleased)

- full flake8/isort compatibility

2.4.2 3.1 (2019-01-20)

- don't encode attributes when wrapping an LDAPUser if they are flagged binary

2.4.3 3.0 (2018-05-21)

- Zope 4 compatibility
- merge and fix old HelpSys API docs into interfaces and add Sphinx doc
- add instance scripts as a test/development convenience
- unbreak saving of bind passwords on the Configure tab
- remove old _SharedResource code and simplify caching
- replace Products.LDAPUserFolder.SimpleCache.SimpleCache with Products.LDAPUserFolder.cache.UserCache, based on dataflake.cache.timeout.TimeoutCache
- replace Products.LDAPUserFolder.SimpleCache.SharedObject with dataflake.cache.simple.SimpleCache
- flake8 whitespace cleanup
- moved the code to GitHub
- officially dropped Python 2.6 support, only Python 2.7 is supported.
- moved documentation to Sphinx
- sanitized buildout test script generation to always use the `exportimport` extra and always test the *GenericSetup* export/import support

- Add `tox` configuration to support automated testing on all supported Python versions
- Removed the LDAPUserSatellite code due to severe bit-rot. Please use the PluggableAuthService package in conjunction with LDAPMultiPlugins to gain the same functionality.
- Removed the CMF tools, please use the package `Products.CMFLDAP` (see http://pypi.python.org/pypi/Products.CMFLDAP/1.0) instead.
- ensure bind passwords used for the LDAP delegate and the user folder do not get out of sync
- Refactor some definitions in the utils module to make them easier to override (Patch by Godefroid Chapelle)
- Fixed a missing string conversion in getGroupedUsers (Patch by Godefroid Chapelle)
- Fix python-ldap error when receiving sets instead of lists for attributes to search on (Patch by Godefroid Chapelle)
- When comparing a login value to login values found on the LDAP server strip the login value first. This follows OpenLDAP behavior which considers values as matches even with trailing or leading spaces in the value query filter. (https://bugs.launchpad.net/bugs/1060080)
- LDAPDelegate: When using a user from the Zope security machinery for the purpose of finding a suitable bind DN and password for connecting to a LDAP server, discard it when it's not been created as the result of a real login and thus has an invalid password (https://bugs.launchpad.net/bugs/1060112)

2.4.4 2.23 (2012-04-23)

- Add `setuptools-git` to `setup_requires` to prevent missing files in the egg release - versions 2.22 and 2.21 will not build due to a missing `VERSION.txt`.

2.4.5 2.22 (2012-04-23)

- factored some tests into separate modules to increase maintainability
- Moved all documentary text files into the egg root

2.4.6 2.21 (2012-04-21)

- Make sure to raise OverflowError if no users can be found when calling `getUserNames` (https://bugs.launchpad.net/bugs/972408)
- switch to using the standalone `dataflake.fakeldap` package for unit tests

2.4.7 2.20 (2011-05-04)

- Fix for CVE-2010-2944 (http://secunia.com/advisories/cve_reference/CVE-2010-2944/), which was never reported upstream by the Debian people, who found the problem 8 months ago (see http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=593466). Thanks guys.

2.4.8 2.19 (2011-01-10)

- Add attribute name to the negative_cache_key so requests for same value but different attribute do not poison the cache. (https://bugs.launchpad.net/bugs/695821)

- The changed base classes in Zope 2.13 did not define isPrincipiaFolderish, so the user folder would no longer show up in the left hand navigation pane in the ZMI. (https://bugs.launchpad.net/bugs/693315)
- Fixed a faulty check for unicode so user expiration will not fail if a unicode value is passed in. Changed all checks for string and unicode to use basestring. (https://bugs.launchpad.net/bugs/700071)
- Fixed an export/import test error so all tests run again.
- The Manager DN Password value on the `Configure` tab in the ZMI showed up in clear text when viewing the HTML source for the rendered page. (https://bugs.launchpad.net/bugs/664976)

2.4.9 2.18 (2010-07-29)

- Added a new flag `purge` to the `ldap-servers` and `ldap-schema` export/import XML elements for finer-grained control over value purging for those two settings if the global purge flag is not set. (https://bugs.launchpad.net/bugs/586970)
- The export/import code did not handle server definitions using the `ldapi` protocol correctly. (part of https://bugs.launchpad.net/bugs/586970)
- When adding a new server definition, the comparison to avoid duplicate server definitions was faulty. Furthermore, operations and connection timeout values were disregarded for duplicate server definitions. (https://bugs.launchpad.net/bugs/586967)

2.4.10 2.17 (2010-05-28)

- Added GenericSetup magic to fully provide the INode interface for the exporter and importer classes, making it easier to nest within other importers. (https://bugs.launchpad.net/bugs/586531)

2.4.11 2.16 (2010-04-15)

- depend on dataflake.ldapconnection so tests run without any hassle.
- Use sha hexdigests instead of digests to build cache keys, digest values can contain non-ASCII characters.

2.4.12 2.15 (2010-04-12)

- Changed import/export test code to be compatible with GenericSetup 1.5
- No longer force-inject the "fakeldap" module into the module namespace sys.modules as "ldap" for testing

2.4.13 2.14 (2009-12-22)

- Updated compatibility with CMF 2.1 and the upcoming dataflake.ldapconnection 1.0
- Potential Bug: Avoid cache hash clashes by recomputing the hash when the admin clears the caches in the ZMI (http://www.dataflake.org/tracker/issue_00629)
- Bug: _lookupuserbyattr created its own user search filter and did not take the _extra_user_filter attribute into account (http://www.dataflake.org/tracker/issue_00640)

2.4.14 2.13 (2009-05-02)

- Factoring: Removed the SSHA module in favor of using Zope's AccessControl.AuthEncoding module to handle password creation for most types of encryption.
- Bug: Binary attribute handling in manage_addUser was broken. Only the first character in the binary attribute's value would be stored.
- Miscellaneous: Weed out all LDAP search calls that would indiscriminately pull all attributes from a user record. This will reduce server load if the user record contains large attributes such as jpegPhoto.
- Feature: Added GenericSetup support with import/export steps for the LDAPUserFolder. When installing the LDAPUserFolder via Buildout, make sure to specify the extra name "exportimport" to automatically pull the GenericSetup dependency: Products.LDAPUserFolder[exportimport]
- Factoring: GenericSetup profile registration and CMF skin folder registration now moved from code to ZCML. Renamed profile "default" to "cmfldap", that's a more descriptive name. The minimum CMF version required for installing the CMF integration is now 2.1.0, which implies Zope 2.10.4 or later.

2.4.15 2.12 (2008-10-21)

- Bug/LDAPDelegate: Use the canonical explode_dn method when splitting up a DN for escaping its values instead of hand-splitting on ",", which breaks if the DN contains commas in any value. Patch by Russell Sim. This also required cleaning up one test that used an invalid DN format. (http://www.dataflake.org/tracker/issue_00623)
- Factoring: For testing, use the fakeldap module from the dataflake.ldapconnection package instead of maintaining a copy here.
- Factoring: Refactored unit tests to use ZopeTestCase and ZopeLite instead of hand-rolling ZODB connections etc.
- Bug: Added explicit CMFDefault dependency for the CMF-related functions by adding an extras_require in setup.py.
- Bug: Make sure a user is purged from the negative cache when the user is explicitly expired. (http://www.dataflake.org/tracker/issue_00617)

2.4.16 2.11 (2008-08-01)

- Feature: The site administrator may now set an arbitrary LDAP search filter expression that will be applied to all user searches in addition to the default filters. Only those user records matching both the default filter and this arbitrary filter expression will be returned. CAUTION: The filter expression must conform to standard LDAP filter syntax. Setting a wrong value will lock out your users! (http://www.dataflake.org/tracker/issue_00615)
- Factoring: Move the LDAP server configuration off the Configure tab in the ZMI to its own LDAP Servers tab to avoid overcrowding the configuration view even more.
- Bug: The unit tests for the LDAPMemberDataTool and the LDAPMembershipTool did not run due to a faulty import.
- Bug: The ZMI Caches tab erroneously suggested that a cached user's last access time would be recorded and/or updated. This was not the case, it is recorded at user object creation and then never updated. The Caches tab will now reflect the creation time. Since the API to set or query the last access time was not used anywhere it has been removed. (in response to http://www.dataflake.org/tracker/issue_00614 by Stefan Loidl)

2.4.17 2.10 (2008-07-21)

- Bug: Recreating the internal cache hash key inside LDAPUserFolder.__setstate__ can lead to values differring from one thread to the next, leading to unnecessary extra LDAP lookups for values already cached under the original key. (http://www.dataflake.org/tracker/issue_00608 by Stefan Loidl)
- Factoring: LDAPUserFolder.__setstate__: Removed old backwards-compatibility gyrations.
- Bug: FakeLDAP could not handle BASE-scoped searches
- Bug: LDAPUserFolder.searchUsers mishandled searches on DN by not passing the correct BASE search scope through. Found by Nico Grubert.

2.4.18 2.9 (2008-06-04)

- Bug: LDAPUserFolder.getUserByAttr: The negative login cache used for preventing repeated LDAP requests when a user enters wrong creadentials was keyed on user login alone. This would prevent subsequent logins with the correct password. Thanks to Tarek Ziade for test and patch and Gilles Lenfant for filing the issue. (http://www.dataflake.org/tracker/issue_00605)
- Refactoring: test suite: Rearrange imports to prevent error messages when the CMF is not present.
- Bug: LDAPDelegate.search: Improve searches on binary attributes such as objectGUID by introducing a method argument that prevents UTF*-conversion of the filter expression passed in. (http://www.dataflake.org/tracker/issue_00576 by Wichert Akkerman)
- Feature: Improve binary attribute handlng by introducing a binary flag for LDAP schema items that is consulted when inserting/modifying an attribute flagged that way. Introduce a hardcoded list of binary attributes to no convert from UTF-8 when searching. (http://www.dataflake.org/tracker/issue_00598 Dragos Chirila)
- Bug: LDAPUserFolder.getUserByAttr: made login attribute and uid attribute retrieval safer by explicitly providing a default. (http://www.dataflake.org/tracker/issue_00602 by Martin Gfeller)
- Bug: ZMI Groups tab: Asking for the type of group via a separate LDAP search for every group listed is unfeasible for installations with large numbers of groups, it is now only done if the total number of groups is less than 50.

2.4.19 2.9-beta (2008-01-01)

NOTE: In order to use the LDAP-based CMF membership components you need CMF version 2.1.0 or higher.

- Bug: Added a __setstate__ hook for deleting old-style logger instances which were removed for version 2.7 but are now showing up as "broken" objects and may prevent Plone migration scripts from working correctly, pointed out by Martijn Pieters. (http://www.dataflake.org/tracker/issue_00574)
- Bug: Removed failing unit test for old-style Zope 2 interfaces that no longer exist in the CMF
- Bug: CMFLDAP skins: Cleanups and changes to align the custom skin scripts and templates with their CMF 2.1.0 counterparts
- Bug: LDAPMemberDataTool: The "Member Properties" ZMI tab was broken due to a typo in the ZPT code.
- Bug: LDAPMemberDataTool: Adjusted wrapUser to match the changed behavior in CMF 2.1.0 and up.
- Bug: LDAPMembershipTool/LDAPMemberDataTool: Since the core CMF tools no longer support the IActionProvider interface the tests to prove the LDAP-based versions support these interfaces have been removed.
- Bug: The functional test rig setup has been changed to avoid DeprecationWarning-Messages from GenericSetup 1.3 and up.

- Bug: LDAPUserFolder.searchGroups: Make the code more defensive for situations where a search would return groups without members, suggested by Nick Davis. (http://www.dataflake.org/tracker/issue_00584)
- Feature: Added negative caching for users to avoid querying the LDAP server again and again for invalid logins. Patch provided by Wichert Akkerman. (http://www.dataflake.org/tracker/issue_00572)
- Feature: added a group/membership mapping for group type "univentionGroup" (http://www.dataflake.org/tracker/issue_00569)
- Documentation: Noted the danger of trying to install the CMFLDAP extensions into a Plone site: Just don't do it, you will suffer!

2.4.20 2.8 (2007-06-13)

NOTE: In order to use the LDAP-based CMF membership components you need CMF version 1.6 or higher.

- ensure CMF tool unit tests run against CMF 1.6 and up
- cmfldap skins: Replace the non-working skin scripts with a set based on CMF 1.6.
- LDAPUserFolder/LDAPDelegate: Change the hash key generation which produces the keys used for caching to use random numbers instead of time-based hashes. LDAPUserFolder will also generate a new hash key whenever Zope is restarted. (http://www.dataflake.org/tracker/issue_00535)
- LDAPDelegate._connect: We now check to see if a new requested connection is known to our configuration by checking the connection string against the saved server information in order to prevent reusing connections instantiated while handling ldap.REFERRAL exceptions, spotted by Riccardo Lemmi. (http://www.dataflake.org/tracker/issue_00548)
- utils: _verifyUnicode was faulty and would return non-unicode if the input was not simple ascii or a unicode object. Kudos for the discovery to Godefroid Chapelle.
- LDAPUserFolder.searchUsers: Faulty code would cause an exception if the LDAP delegate returned a failed search with exception information, discovered by Andreas Gabriel.
- LDAPUserFolder.getGroupDetails: Removed a hardcoded list of possible group member attributes and replaced it with utils.GROUP_MEMBER_MAP, which is used anywhere else. Good spot from Helge Tesdal. (http://www.dataflake.org/tracker/issue_00560)
- LDAPUserSatellite.getGroups and LDAPUserSatellite.getAdditionalRoles: Replace occurrences of hardcoded group member attributes with usage of utils.GROUP_MEMBER_MAP, also suggested by Helge Tesdal.
- Removed Zope 2.7 compatibility code and cleaned up imports
- Removed compatibility code for CMF < 1.6

2.4.21 2.8-beta (2006-10-16)

NOTE: The python-ldap requirement is now version 2.0.6 or higher

- Fixed a broken security declaration for searchGroups and a left-over form tag in the Users tab (thanks to Klaus Barthelmann)
- LDAPDelegate.modify would attempt to modify a LDAP record even if the list of modifications was empty. This is now logged without any further call to python-ldap to prevent some servers from throwing UNWILLING_TO_PERFOM. (http://www.dataflake.org/tracker/issue_00528)
- DNs were not properly escaped for such edge cases that needed escaping, like values starting with "#". (http://www.dataflake.org/tracker/issue_00507)

- Changes in GenericSetup meant attempting to register the extension profile for installing the CMFLDAP tools would fail in CMF >= 1.6.
- Group deletion for groups with non-ASCII and non-UTF8 characters was broken, discovered by Eric Brun (http://www.dataflake.org/tracker/issue_00527)
- Unforeseen software combinations, such as CMF < 1.6 in combination with GenericSetup could prevent Zope from starting up because the LDAPUserFolder initialization module would throw an error.

2.4.22 2.7 (2006-07-20)

- Sidnei da Silva took the time to root out any use of mutable variables in method argument lists.
- Completely refactored the way searches are handled by the FakeLDAP testing fixture. The new code uses intelligent parsing to make sense of a query and apply it in a generic way instead of trying to sniff a filter to guess where the query came from and what the query was attempting to do. Kudos for a whole bunch of time spent go to Sidnei da Silva.
- Added a more powerful groups search method named searchGroups to improve group searching capabilities for e.g. Plone and PlonePAS. Wichert Akkerman provided code and tests.
- Added a more powerful user search method named searchUsers. Unlike findUser, searchUsers allows for more than one attribute to be searched on. findUser has been reduced to a simple wrapper around searchUsers. My thanks for inspiration and an initial implementation suggestion go to Wichert Akkerman.
- Enabled utils.to_utf8 to handle unicode (continuing work on http://www.dataflake.org/tracker/issue_00480)
- The use of zLOG, and the SimpleLog module, have been removed in favor of using the Python logging module throughout.
- Software dependencies are now listed concisely in one place, a new DEPENDENCIES.txt file.

2.4.23 2.7-beta (2006-03-02)

- Harden SimpleLog against cases where the log message included strings like %s, caught by Wichert Akkerman. (http://www.dataflake.org/tracker/issue_00491)
- In ActiveDirectory, it is possible to have records (specifically internal system accounts) that have the correct objectClasses to qualify as user records, but they lack the attribute designated as the chosen UID attribute. Thanks to Wichert Akkerman, these are now disregarded. (http://www.dataflake.org/tracker/issue_00484)
- Make sure objectGUID, when set on the LDAPUser as a property, gets treated specially (discovered by Wichert Akkerman in the course of clarifying http://www.dataflake.org/tracker/issue_00480)
- The SimpleLog.zLOGLogger log method ignored the `args` parameter (http://www.dataflake.org/tracker/issue_00474, thanks go to Mark Hammond)
- Repaired warings appearing in Zope 2.8.5 due to a couple typos in security declarations.
- Fix breakage when local groups storage is used and no groups are assigned to a user: When roles are changed to another empty value an error occurred (http://www.dataflake.org/tracker/issue_00478 by Junyong Pan)

2.4.24 2.6 (2005-10-29)

- Expanded findUser with an argument "exact_match" to signal whether a search term passed in should only return exact matches or wildcard matches. This is also exposed on the Users tab in the ZMI as a selection widget for the search form. (Inspired by a suggestion from Sidnei da Silva)

2.4.25 2.6beta3 (2005-09-22)

- Folded the CMFLDAP product into the LDAPUserFolder package
- Revamped the unit tests to share test fixture creation code and to work in both Zope 2.7 and 2.8 without problems.
- The LDAPUserFolder factory method and the initialization code were massively simplified. A lot of duplicated code was removed. When adding a LDAPUserFolder, there is no longer a separate Add view. The user folder will be created straight away and the admin will be redirected to the Configure tab of the new instance. **Note**: If you have code that programmatically instantiates LDAPUserFolder instances then you must change it. See the unit test files "setUp" method for an example how to do it from this point on.
- Fixed a serious bug that crept into version 2.6beta1 and which led to users being able to log in with a wrong password or no password.
- The getId method on the LDAPUser class neglected to encode the user ID to an encoded string and handed back unicode, which could lead to strange failures elsewhere. Most code handling IDs is not equipped to deal with unicode.
- A bug had crept into the logging subsystem that could cause spurious error messages. (http://www.dataflake.org/tracker/issue_00462)
- The user records found via the Users tab search were not consistent with the users that can actually log in because the search on the Users tab did not filter out records that do not match the user object classes as defined on the Configure tab. (http://www.dataflake.org/tracker/issue_00260 and http://www.dataflake.org/tracker/issue_00445)

2.4.26 2.6beta2 (2005-07-28)

- Previous changes in how the LDAPUserFolder handles the conversion of LDAP group memberships to Zope roles (it was made explicit as opposed to automatic and implicit) made the LDAPUserSatellite less useful for users who expected LDAP group names to automatically show on the user object. Now the LDAP User will carry a hidden field for all current LDAP group memberships, which can then be consulted by the LDAPUserSatellite to determine what additional roles to hand out. (Suggestion by Dirk Datzert)
- The LDAPUserSatellite configuration screen would blow up trying to determine the logging level, which has been removed.
- Before returning a new connection in the internal LDAPDelegate connection methods the Manage DSA IT control was enabled. This was the result of misunderstanding the control - it really is only needed to directly access and manipulate a referral or alias entry without having the server send you to the referred or aliased server.
- The old behavior of mapping every LDAP group name a user is member of to a Zope role of the same name can now be reactivated using a new configuration option named "Group mapping" on the Configuration tab. Many thanks to Dirk Bergstrom for a set of patches and unit tests. (http://www.dataflake.org/tracker/issue_00459)

2.4.27 2.6beta1 (2005-07-05)

- Spell out how to safely upgrade in README.txt by using the emergency user to delete/recreate the instances.
- Made the getAttributesOfAllObjects method more resilient by always providing a key per queried attribute in the resultset (http://www.dataflake.org/tracker/issue_00456 by Pierre-Julien Grizel)
- Applied a similar fix to getUserIds and getUserIdsAndNames that was applied for Tracker issue 441 to make sure empty resultsets don't lead to catastrophic failures (http://www.dataflake.org/tracker/issue_00446 by Pierre-Julien Grizel)

- An earlier special-casing applied by Chris McDonough to correctly handle AD objectGUID values has been applied in a second place, in the findUser method (patch by Mark Hammond).
- Deleting a user record would be short-circuited if the user record itself was not in the DIT anymore, e.g. because someone manipulated the DIT without the user folder knowing about it. This prevented cleanups for group memberships to be performed. (http://www.dataflake.org/tracker/issue_00439 by Hans-Juergen Sell)
- The getUserNames function did not react correctly in the face of an empty resultset from getAttributesOfAllObjects and would prevent admins from using the ZMI local role management view. getUserNames now also raises a OverflowError if no results have been returned in order to show a simple text input widget on the local role management view instead of the multiple choice select box. (http://www.dataflake.org/tracker/issue_00442 by Andrew Veitch and http://www.dataflake.org/tracker/issue_00441 by Hans-Juergen Sell)
- Added the new logging machinery to the LDAPDelegate class which improves lower-level LDAP problem discovery.
- Moved away from the current way of logging to a purely zLOG-based mechanism. This will make sure that all logging for Zope is in one and the same place and that more information can be passed along to the logging mechanism, such as tracebacks. (http://www.dataflake.org/tracker/issue_00438 by Mark Hammond)
- Refactored the code that has python-ldap dependencies so that only the LDAPDelegate instance now holds all the cards. This enables plugging in different delegate implementations because subclassing LDAPDelegate and overriding implementation details has become easier. (http://www.dataflake.org/tracker/issue_00438 by Mark Hammond)
- Added a registry for delegate implementations so that other delegate classes can register themselves with this registry and become available to the LDAPUserFolder during instantiation.

2.4.28 2.5 (2005-04-16)

- Make the error message that gets created when a connection to the LDAP server fails a tick more verbose
- Remove an optimization that would cache unsuccessful lookups in order to prevent undue strain on the LDAP server. The cached records would prevent a LDAP server lookup for a pretermined time. This turns into a problem where code tried to check for the existence of a user before adding it and then trying to retrieve the new user to operate on it. Since the first lookup will have created an entry in the cache the second lookup to retrieve the user will always return None, even though the user might have been added successfully.

2.4.29 2.5beta3 (2005-04-11)

- Using the full DN as the user's ID was broken since the AD-related "objectGUID" changes in 2.5beta1 due to a broken "if" statement.
- Replace deprecated usage of ldap.is_ldap_url, thanks to Sebastien Munch (http://www.dataflake.org/tracker/issue_00419)
- Add caching to getUserById and getUserByDN, it got "lost" during the cache changes introduced for version 2.4 (http://www.dataflake.org/tracker/issue_00402)
- Removed the test_all.py helper script - the only supported way to run the unit tests is using "zopectl test" under Zope 2.7.x and up

2.4.30 2.5beta2 (2005-01-23)

- Expiring users from the cache did not work correctly when a user password was changed or when the roles were edited and the user's DN contained non-ASCII characters, reported by Helge Tesdal. (http://www.dataflake.org/

tracker/issue_00409)

- In addition to the network-related timeout feature introduced on 2.5beta1 there is now a operations timeout, which is useful if you have to live with strange network conditions that drop the connection between the LDAPUserFolder and the LDAP server without the LDAPUserFolder knowing about it.
- The LDAP over IPC protocol can now be used to communicate with the LDAP server through a file socket. Please see the README for additional notes on LDAP over IPC.

2.4.31 2.5beta1 (2004-11-20)

- The setting for groups storage was not carried over from the Add screen when instantiating a new LDAPUserFolder. http://www.dataflake.org/tracker/issue_00387 by Pierre-Julien Grizel.
- The getAttributesOfAllObjects method promised to return a mapping but returned an empty list in case of errors.
- Ignore "DN" when passed in as an attribute to modify within LDAPDelegate.modify (it is not possible to modify a user's DN this way).
- When changing user record attributes the "multivalued" flag from the LDAP Schema configuration was never consulted and if the new value contained a semicolon (;), it would automatically be considered multivalued. This made it impossible to have single-valued attributes with semicolons in it. (http://www.dataflake.org/tracker/issue_00395)
- Revamp tests so that they can be run comfortably using the Zope 2.7.3+ idiom of running via "zopectl test".
- Deal transparently with marshalling ActiveDirectory "objectGUID" values. These are binary values, so they can't be sent without marshalling across the network. This makes it possible to use an AD objectGUID a User Id attribute,
- Added a new "Network Timeout" setting to the LDAP server configuration. The Network Timeout prevents the LDAP connection from hanging indefinitely if the network connection cannot be established and connection attempts do not raise an immediate connection error. Important note: It is possible that during a request several attempts at connecting to the LDAP server are made. The time it takes for the LDAPUserFolder to return control to Zope will be the sum of the connection attempts multiplied by the chosen Timeout value.

2.4.32 2.4 (2004-07-31)

- Small fix to enable non-ASCII characters in LDAP group names (JTracker issue 381 by Andreas Jung)

2.4.33 2.4beta3 (2004-06-11)

IMPORTANT NOTE: This version of the LDAPUserFolder does away with the old behavior of implicitly mapping LDAP groups to Zope roles. Any Zope roles that get conferred are governed by the "LDAP group to Zope role" form on the "Groups" tab. If you relied on this behavior please create the appropriate mappings in your instance.

This version introduces a switchover to the new Zope Public License (ZPL) version 2.1, which will bring the LDAPUserFolder in line with future Zope releases.

- Added a method to retrieve the URI for the currently active LDAP server connection which is now shown in the LDAP Servers part of the Configure tab.
- Added MD5 to the list of available default password encryption methods
- Refactored caching using a new simple cache class contributed by Chris McDonough.

- getAttributeOfAllUsers method removed in favor of a more general getAttributesOfAllObjects method on LDAPUserFolder class. Other methods that deal with mass query of object attributes should likely be gradually refactored in terms of this method at some point, but for now there is some duality in the way attribute-centric object queries are done.
- getUserIds method results are now cached.
- API addition: getUserIdsAndNames method added to LDAPUserFolder, which returns a sequence of two-tuples (id, username) for each user found in the tree. This result is cached.
- Added minimal support for setups where user and groups base DN is actually the same subtree (e.g. ActiveDirectory). LUF now specifies a filter for LUF-specified user object classes during some calls dealing with searching for users (getUserIds, getUserNames, getUserIdsAndNames). **If you were relying on older behavior where all objects in a tree are returned as users from these calls regardless of their actual object class, you will now need ensure that you appropriately specify your user object classes on the main configuration page.** One notable exception to this rule is that searching for from the LUF "Users" tab will still expose groups in search results when user and group base DNs are the same. This is considered a minor bug in the case that a set of user object classes are specified and should be fixed.
- The password is no longer logged when Debug-Level logging is enabled. To go back to the old behavior the old log code is still in place, but must be enabled by hand in the python code. This represents a reversed decision on JTracker issue 247.
- Refactored the Groups tab in the Zope Management Interface (ZMI) to be less cluttered and be clearer about the difference between group records in LDAP versus roles in Zope.
- LDAPDelegate's search method now ignores nonstandard internal referrals returned by ActiveDirectory when querying it against port 389. These referrals aren't returned when querying against AD's global catalog port, so they seem safe to ignore. This may make it possible to use LUF against the normal LDAP port (389) of an AD server without needing to fall back to the GC port.
- Small optimization: when the login id is a DN, don't bother attempting to contact the LDAP server when the login id isn't a valid DN. This shortcuts the possibility that the LDAP server will be queried needlessly for names that aren't real DNs (like those for users in user folders defined above the folder in which LUF resides).
- LDAP groups are no longer implicitly mapped to Zope roles. The roles that are visible on user objects created by the LDAPUserFolder are dependent on the "LDAP group to Zope role" mapping that can be manipulated on the "Groups" tab in the ZMI. The existing behavior of adding the roles specified as "Default user roles" on the "Configure" tab to all authenticated users remains the same. This change means that the administrator now has *full control* over what roles a user can have.

2.4.34 2.4beta2 (2004-04-14)

From this version on the LDAPUserFolder product will drop compatibility with Python 2.1. You should use Python 2.2.3 with Zope 2.6.x or Python 2.3.3 with Zope 2.7.x

The separation of Login and User ID as described below is only fully supported with Zope versions *higher than 2.7.0*. For earlier version you should select the same attributes for both Login and User ID.

Kudos to Chris McDonough to check in the changes below!

- The following API methods of LDAPUserFolder and LDAPDelegate did not work properly when they were passed a unicode object (as opposed to a string) as one of their arguments:

  LDAPUserFolder.manage_edit LDAPDelegate.edit LDAPDelegate.insert
- When selecting the full DN as login attributes a user was unable to log in if the DN contained non-ASCII characters (JTracker issue 372 by Ralf Herold).

- Distinction between user id and login name. You can now configure the attribute used for a user id to use a different LDAP attribute than the attribute used for a user's login name. This allows you to configure LDAPUserFolder, and thus Zope, to use an identifier other than the user id as a login name. This feature is useful if you wish to use email addresses or other identifiers which may change over time as login names. If you employ this feature, you may allow login names to change (by updating the LDAP attributes of the entries to which they refer), but Zope security depends on the user id remaining constant; you still may not allow the attribute used for the user id to change without performing "surgery" on your Zope instance to update local role maps stored in your ZODB and so forth. For backwards compatibility purposes, it is possible to set the user id attribute and the login name attribute to point to the same LDAP attribute. In the common case, users upgrading from older LDAPUserFolder versions, whom typically employ "cn" as their login name attribute should also employ "cn" as their user id attribute. The addition of this feature has caused some method signatures to change in a non-backwards-compatible way. These are LDAPUserFolder.manage_addLDAPUserFolder, LDAPUserFolder.LDAPUserFolder.__init__, and LDAPUserFolder.LDAPUserFolder.manage_edit. Additionally, code which relies on LDAPUserFolder's "getUserById" returning the same value as its "getUser" method will break as a result of this change. There is a new API method exposed on the LDAPUserFolder object as a result of this change: "getUserIds", which returns an enumeration of all user ids.

2.4.35 2.4beta1 (2004-03-23)

This version *requires* you to delete and reinstantiate all existing LDAPUserFolder and LDAPUserSatellite instances!

- Added call to clear the internal caches after a user has been added so that getUserNames cannot return a stale user list (JTracker issue 362 posted by Nick Bower).
- The getMultiValuedUserAttrs method was protected by an invalid permission (JTracker issue 355 by Florent Guillaume).
- Add shortcut in getUser to immediately return None is the passed name is empty or None.
- If it is available I am now using the ReconnectLDAPObject for LDAP connections since it promises to hide temporary connection problems and long connection timeouts. This could potentially fix JTracker issue 324 by P.-J Grizel.
- A new SharedResource module based on Dieter Maurer's package is now used to provide storage for resources that benefit from being globally unique such as caches, the log and the LDAP server connection object. This brings several advantages, such as:
  - The log tab will always show the same thing, regardless of which Zope thread handled the rendering.
  - The LDAP connection itself does not need to be estableshed for each thread, one connection handles all threads. This will probably fix JTracker issue 337 by Florent Guillaume.
  - The user object cache is globally unique now, meaning the number of trips back to the LDAP server should be reduced.
  - The list of user IDs generated by calls to getUserList is no longer a thread-level variable but globally shared, meaning this potentially expensive search operation will be performed less often.
  - These changes were also applied to the LDAPUserSatellite log, user to role mapping and expiration mappings.

2.4.36 2.3 (2003-12-18)

- Noticed that sometimes "empty" authentication credentials lead to unnecessary lookups for non-existing users. Relaxed a specific authentication check so this is prevented.
- The unicode changes had possible disabling consequences for group-to-role mappings defined on the Groups tab. Thanks go to Helge Tesdal for pointing that one out.

2.4.37 2.3beta3 (2003-11-30)

- Fixed a couple buglets found by Florent Guillaume (JTracker issue 333).
- Florent also noticed code that would trigger unnecessary MODRDN calls when a user record was updated. This extra call did not damage the record, it was just unnecessary work (issue 334).
- Dieter Maurer provided the explanation for a recursion error in the __getattr__ method on the LDAPUser object that a few people had run into (JTracker issue 338 by Michael Crawford).
- The getGroupedUsers method was not working if the groups are stored in the user folder itself (JTracker issue 342, thanks Florent Guillaume again).
- Spurred by Helge Tesdal and Nate Aune I spent a little more time on the unicode-ability. Now a user that has non-ASCII characters not just in arbitrary attributes but also in attributes that form part of the full DN are processed correctly. This required quite a few changes, so any feedback is very welcome.

2.4.38 2.3beta2 (2003-11-02)

- Cut down on the number of LDAP lookups in cases where the user lookup happens "anonymously", meaning not as part of a normal authenticated request but from the Zope security machinery for things like ownership-related security checks. Thanks to Kyler Laird for bringing this one up.
- All user lookups are now limited to those object classed defined in the "User object classes" configuration setting on the "Configure" tab. Previously the lookup policy was much more lenient and accepted every record where the login matched - now the object classes have to match as well. ***WARNING - THIS MIGHT BREAK YOUR SITE IF YOU WERE SLOPPY WITH THE OBJECT CLASSES SETTING AND USAGE!*** Due to the possible breakage I had been sitting on Tracker issue 294, filed by Andy Dustman, for quite a while before going with it. Thanks for keeping the pressure on - it is "the right thing" to do.
- The "Users" tab will now show a little more information on the user record detail view by default, namely the DN and the object classes.
- The unit tests have been changed to work with the latest and greatest (Zope 2.7 and Python 2.3.2), which is now the default platform used to test and develop this product.

2.4.39 2.3beta1 (2003-09-29)

- Use of the distinguished name as login attribute was broken in version 2.2 - thanks to Ralf Herold for the information (JTracker issue 312)
- The API documentation for manage_addUser in the Zope Help System was slightly off, thanks go to Eugene Prigorodov for pointing that out (Issue 319).
- Cleaned up LDAP filter strings used by the product to have surrounding parentheses.
- Enable correct handling of DN elements that contain bad characters, such as backslash-escaped commas (Bug report by Stephen Kirby)

2.4.40 2.2 (2003-08-08)

- User attributes can now be declared "multi-valued" in the LDAP Schema, thereby ensuring that all values for that attribute are stored on the user object (Feature request by Jean Jordaan, JTracker issue 294).
- While investigating JTracker issue 309 ("problem changing password") it became apparent that previous fixes to correctly use mapped attributes during user creation were flawed. Also, _expireUser is now more resilient against receiving invalid user information.

2.4.41 2.2beta4 (2003-07-24)

- LDAP Referrals are now chased for searches as well. (JTracker issue 277 by Eric Brun) LDAP Referrals in general *require* LDAP server support for version 3 of the LDAP protocol. Almost all newer servers should be able to handle that.
- Removed non-existent "_expire" call from the interfaces file for the LDAPUser class (JTracker issue 303 filed by Jean Jordaan)
- Added "clear" password encryption scheme to the choices available when adding a new LDAPUserFolder (JTracker issue 295, thanks to Andy Dustman)
- Added some (obviously missing) logging calls. Thanks to Jean Jordaan for telling me about it (JTracker issue 300). Also, added a missing message return from the LDAPDelegate modify method.
- Revamped group handling a little bit so that the GROUP_MEMBER_MAP mapping in the utils module is the central place where permissible groups and their member types are stored. Fixed issue 289 by Eric Brun which was suffering from a related problem at the same time.
- If a new user is created and the form fields are not named after the real LDAP attribute names but with mapped names as specified on the LDAP schema tab the correct reverse translation will now be done (JTracker issue 301, thanks to Doug Winter)
- Cleaned out a bunch of unneeded imports
- Added some very interesting ActiveDirectory secrets uncovered by Larry Prikockis to the ActiveDirectory README. This has the potential of helping a lot of people who have difficulties integrating Zope and ActiveDirectory.

2.4.42 2.2beta3 (2003-06-06)

- The routine used to create a crypt-style password string did not take enough precautions to ensure that the salt value used stays pure ASCII. This could prevent users from logging in.

2.4.43 2.2beta2 (2003-05-14)

- The list of LDAP servers will now respect the order in which they were added and the connection process will go through the servers in that same order, starting at the top of the list as visible on the "Configure" tab. (JTracker issue 284 by Dirk Datzert)
- Started a separate README for those hapless users who are stuck on Active Directory with input from Philipp Kutter (JTracker issue 280), see README.ActiveDirectory.txt
- If roles were stored locally and a user with locally stored roles had all roles removed that user would still show up in the user listing, even if the user record itself was removed from LDAP. Now removal of all roles will clean the internal roles storage mechanism correctly. Thanks go to Hans-Juergen Sell for letting me know.
- When a user logs in the application will no longer construct the user object with the name typed in by the user but will look it up in the LDAP record itself. That way a user will always be represented by the same username, regardless of what capitalization was used upon login (JTracker issue 282, thanks go to Ronan Amicel)
- Domain restrictions put on the emergency/init-users were not respected, thanks to Dirk Datzert for pointing that out in JTracker issue 283.
- Broke the Caches tab if and when the anonymous cache contained any users, the display for anonymous cache users was calling a non-existing method. (JTracker issue 281, my thanks go to Ronan Amicel)
- Logic error in getGroups corrected that could lead to binding with an invalid user/password pair. Now the decision what to bind as is left completely up to the LDAPDelegate itself.

- Added workaround for changed behavior of ldap.explode_dn which will blow up now if the passed-in DN does not contain at least one key=value pair.
- Removed superfluous argument to manage_setUserProperty (Tracker issue 270 by Dirk Datzert)
- Fixed manage_setUserProperty errors that crept in during the last great code reorganization and also added a unit test to exercise this method. (Tracker issue 269, thanks to Dirk Datzert again for pointing that out)

2.4.44 2.2beta1 (2003-04-18)

- Cache timeouts can now be set from the Caches tab in the ZMI (Tracker issue 263 by Michael Lindig)
- "Manager DN usage" set to "Always" would still bind as the user itself after the initial bind, now it only uses the Manager credentials. Had to insert a bind as the user to determine password validity, though. (Tracker issue 266)

2.4.45 2.1 (2003-04-14)

No significant changes between 2.1 beta3 and 2.1

2.4.46 2.1beta3 (2003-03-16)

- Fixed a LDAP server misbehavior where a bind operation with a valid user DN but empty password would seemingly succeed. This behavior was only obvious in 2.1beta2 because I removed code I considered obsolete. Also added unittests for authentication and extended the FakeLDAP module to emulate LDAP server binding behavior. (Tracker issue 257, my thanks go to Jan-Wijbrand Kolman)

2.4.47 2.1beta2 (2003-03-02)

- Apparently there are situations when a call to getGroups returns a tuple. Code in the LDAPUserSatallite expected it to be a list (Tracker issue 244).
- If the LDAPUserFolder was configured to always bind using the Manager DN it was possible to log in with the wrong password (Tracker issues 246 and 248, thanks go to Michael Lindig).
- Found a problem deleting all values for a user attribute from the ZMI which would throw an error. Discovered while looking at the (unrelated?) issue 251 in the tracker, which also dealt with a problem when clearing an attribute.

2.4.48 2.1beta1 (2003-02-24)

- Cleaned up a mismatch between the delegate edit method and signature expected by the LDAPUserFolder code that talks to it (Tracker Issue 224 pointed out by Albert Chin-A-Young and others)
- More cleanup in the way a LDAPUserFolder authenticates to the LDAP server. The setting specified under "Manager DN usage" is now respected for all record modifications and deletions as well.
- Michail Bachmann pointed out some code errors in the LDAPUserSatellite code that had crept in when switching to using the LDAPDelegate (tracker issue 233).
- Finally added a full suite of unit tests for most components in the package.
- If your LDAP server hands out referrals during an attempted write operation (add, modify or delete a user record) then this is now handled correctly, at least if you run OpenLDAP and python-ldap versions 2.0 or higher.

- Implemented read-only mode where any writes to the LDAP server are disabled (Tracker issue 228 filed by Tom Deprez).
- Officially removed compatibility with python-ldap 1.x versions. Due to an oversight on my part some incompatible code was already in the 2.0-series, but now I am finally dropping any pretenses about supporting that old version.

2.4.49 2.0 (2002-12-24)

- Made encoding tasks a little cheaper if you set the encoding to UTF-8 in the utils.py "encoding" settings (Tracker issue 203, thanks go to Artur Zaprzala for suggesting a good way of handling this)

2.4.50 2.0beta3 (2002-12-13)

- Changing a password would result in an error (Tracker issue 204 posted by Massimiliano Russo)
- If no results were returned from a user record search inside getUserDetails then a faulty check for the length of the returned result set would make it error out. Tracker issue 206, posted by William Carrel.
- Make sure that rebinding in _lookupuser will really fulfill the promise of "uses the DN and password of the currently logged-in user" in case the selected Manager DN usage calls for it. (Tracker issue 210 by Oliver Pabst)
- "clear" has been added as a choice for password encryption mechanism. The password will be stored unencrypted in LDAP in that case. (Tracker issue 209 by Oliver Pabst)

2.4.51 2.0beta2 (2002-12-06)

- Thanks to a patch from Artur Zaprzala the default encoding that gets applied to results from the LDAP servers can now be changed in the utils module, whereas before it was hardcoded to be "Latin-1" in all places.
- The getUserDetails method API was extended to allow passing in a sequence of desired attributes. This is helpful when certain attributes (e.g. binary pictures) are not needed and would unnecessarily slow down the request. This feature suggested by Artur Zaprzala.
- The module reorganization broke the decoding of UTF-8 values which meant non-ASCII characters appeared as garbage characters. Artur Zaprzala posted a set of patches in Tracker issue 198 to fix the issue and simplify/improve the unicode handling in general.

2.4.52 2.0beta1 (2002-11-30)

Version 2.0 represents a major code base refactoring. The main goals were code simplification, cruft removal and improving maintainability for me. While this meant putting the axe to some features it also enabled me to implement some other functionality that would have been much harder to do using the old code.

- ZBabel support has been discontinued. I have received very little (meaning No) feedback on it and even before it was offered only very few people requested it. I myself did not have an environment set up where I could maintain the translation dictionaries, mainly because the way they are updated is (in my opionion) a huge PITA. I got tired of lugging code along that got more stale with every update I did to the main cod. Since I have been on a simplification spree for version 2.0 it was one of the first items to go. My apologies to Dirk Datzert who performed the most of the ZBabel integration work last year.

- Cookie support is no longer built into the product. If you need cookie-based authentication I recommend installing the CookieCrumbler product alongside the LDAPUserFolder. It performs all functionalities of the built-in cookie support. See http://www.zope.org/Members/hathawsh/CookieCrumbler for information and download.
- You can now specify multiple LDAP servers to be used by the LDAPUserFolder. Servers are used in a failover fashion. If the first server in the list is down the next one is contacted, etc. This assumes that the LDAP data structure on both servers is identical, e.g. the users search base is the same.
- The LDAPUserSatellite can now be used in recursive fashion. This means it can go out and consult all LDAPUserSatellites in its acquisition path and have them make any role manipulations before doing its own work, thereby getting a cumulative effect. Please use caution with this feature because it is potentially very expensive.

2.4.53 1.6 (unknown)

That's it, folks... this is the end of the 1.x line of LDAPUserFolders. The new version, LDAPUserFolder 2.x, is out by now and I encourage everyone to give it a try. It has many added features and, above all, a refactored code base that makes it easier for me to maintain and improve.

From this point on no new features will be added to the 1.x series, and only urgent bug fixes. All development will concentrate on the 2.x series.

2.4.54 1.6beta3 (unknown)

- Fixed a bug that could allow access with an invalid password (Tracker issues 185, 188)
- Some more small logging improvements

2.4.55 1.6beta2 (unknown)

- A brand new object can now be instantiated after upgrading to this version. It's called "LDAPUserSatellite" and is used to manipulate roles for a user based on the context the user is in if a LDAPUserSatellite is around. Roles can be manipulated by applying a mapping from LDAP group to additional Zope role and/or performing a group record lookup in an additional groups search path on your LDAP server. The LDAPUserSatellite does not directly change global roles on the user object like the LDAPRoleTwiddler and LDAPRoleExtender did, it uses internal Zope security mechanisms to compute roles based on context. This new object replaces both the LDAPRoleTwiddler and the LDAPRoleExtender, which are hereby deprecated. Thanks go to Dirk Datzert who did some extensive testing and helped me hunt down a lot of bugs.
- Better logging for using cached users
- Caches ZMI tab more informative by presenting both authenticated and anonymous cache contents
- Update some outdated help files for the LDAPUserFolder

2.4.56 1.6beta1 (unknown)

- Group type accessGroup added to the list of group records recognized and usable within the LDAPUserFolder. Michael Stroeder spotted this type of group on a IBM SecureWay directory server.
- More efficient groups search filter for specific user record, suggested by Michael Stroeder.
- Logging and caching are factored out into instance-level objects
- The security model has seen a complete change to make it simpler and to respect access controls placed on the LDAP server itself more:

  - providing a Manager DN and password is optional
  - if a Manager DN has been provided in the configuration then that DN will be used to bind for every single LDAP operation
  - if no Manager DN has been provided then the current user's DN will be used for binding.
  - if no Manager DN has been provided and a user who authenticated against another user folder is attempting to perform LDAP operations it will be performed with an anonymous bind.

  This all implies that if you want to make changes in LDAP that require specific rights you must either log in as a user with those specific rights or use the less security-conscious workaround of providing a Manager DN in the LDAPUserFolder configuration. If you attempt to make changes with a Manager user authenticated against another user folder you might not be able to, which might be a source of confusion for some Zope admins.

- Catch ldap.PARTIAL_RESULTS after issuing a search request to the server, something the Micro$haft "Active Directory" server seems to like doing. Thanks go to Brad Powell for reporting this nonstandard server behavior.
- Reclassified and clarified some logging calls and their message output.
- A lot of "whitespace normalization" (hate that expression!) and fixes to overly long lines of code.
- Handling of multi-valued attributes has been cleaned up and changed slightly. If an attribute value contains semicolon (;) characters it will be assumed to contain a semicolon- separated list of values. The ZMI "Users" tab will also display semicolon-separated values for all multi-valued attributes when you view the record.
- A misconfigured Users base DN setting is now less likely to lead to complete blowups upon trying to connect to the LDAP server so that access to the container will always remain intact and the LDAPUserFolder can be reconfigured or deleted if needed.
- No blowups from getUser if the name passed in is not a string, just returns None instead now. (Tracker issue 166 filed by Romain Eliot)

2.4.57 1.5 (2002-08-17)

- Due to the way user object caching was implemented local role lookup would break for those users who have logins that are not all lowercase. This has been fixed. (Tracker issue 163, my thanks go to John Hohm who did a lot of the detective work for this one himself)
- Using a better search filter in case getGroups is asked to return all groups available to the LDAPUserFolder. Improved the doc string for getGroups to clarify its usage. Michael Stroeder suggested the better search filter.

2.4.58 1.5beta3 (2002-08-02)

- New method getLocalUsers added to allow for retrieving all user DNs and their roles that have roles stored locally. If user roles are stored locally this method is now used on the Users ZMI tab to show a list of all users with locally stored roles. This is more or less a convenience so that the admin does not have to search for a specific record and go into the detail screen to find out about a user's roles.
- The implementation for getGroupDetails was incomplete for locally stored groups. It is now fully implemented.
- New method "getGroupedUsers" will return a sequence of user objects for the groups you pass as argument. If no groups are passed then user objects from all groups that are visible to the LDAPUserFolder are returned.
- Make unwrapped LDAPUser objects a little more useful by ensuring __getattr__ can now find the DN attribute. Trying to call getUserDN on a unwrapped user object will always raise an error due to the nature of wrapping and security declarations. __getattr__ does not raise this error.
- manage_setUserProperty is now more useful by allowing set set empty properties, which it did not before. (Tracker Issue 158, thanks to Sven Thomsen)

- The manage_editUser method will no longer blow up if the specific user's RDN attribute is not part of the values passed in. It will now simply take the old record's RDN value instead.

2.4.59 1.5beta2 (2002-07-08)

- The latest versions of OpenLDAP seem to complain about the LDAP protocol in use if it is not LDAPv3. Added a workaround that catches the complaint and explicitly sets the protocol.
- Corrected some faulty default arguments that could have caused errors in certain cases.
- The group search scope was mis-applied to a search that takes a group DN and returns its objectClass. This would cause errors if SCOPE_ONELEVEL is the groups search scope because that scope does not include the object pointed to by the group DN. Changed to always use SCOPE_BASE (this scope searches the current object only) instead (Tracker issue 141, thanks go to Philippe May).
- A similar bug as the one above afflicted the _lookupuser method. Changed search scope to SCOPE_BASE as well. Derrick Hudson spotted the problem.
- Added workaround for a (supposed) shortcoming in python-ldap where a DN is not part of the search results dictionary even if asked for it explicitly. Also found by Derrick Hudson.

2.4.60 1.5beta1 (2002-05-30)

- Small fix on add form to ensure form element naming is consistent (Tracker issue 139 by David Riggs).
- Instead of adding a workaround for the (faulty) ability to create and have empty group records on Netscape directory server products (which then won't show up on the LDAPUserFolder "Groups" tab) I have added a paragraph in the README that addresses why it happens and what to do.
- A stupid syntax error on my part prevented the "SERVER_DOWN" exception that was used to determine the freshness of a reused connection object to ever be caught correctly. Brad Powell pushed my nose into that and made me fix it.

2.4.61 1.4 (2002-05-17)

- All actions performed on the management tabs with the lone exception of the "Custom Forms" tab will now go back to the same tab, with the correct tab highlighted. (Tracker issue 127, thanks to David Riggs)
- Expiring users out of the caches when the record got changed was not working in all cases. The expiration is now more explicit and involves manipulating the caches directly instead of changing the expiration time on the user object (Tracker item 128).
- IE on windoze misbehaves when setting a cookie where expiration is set to an empty string. All other browsers (surprise surprise!) behave correctly, but IE will foil any login attempts when using cookie mode. Added a workaround (Tracker issue 129).

2.4.62 1.4beta3 (2002-05-08)

- The code used to format exceptions in the utils method was called the wrong way and failed when it was called to format an exception (Tracker issue 125, thanks to David Riggs).
- A juxtaposition in arguments to manage_edit led to segfaults in some applications (spotted by Tres Seaver).
- Logging is extracted into its own module (SimpleLog), thereby making it easier to extend later.

2.4.63 1.4beta2 (2002-05-07)

- A small bug crept in while making the user cache case- insensitive (Tracker # 123 from Jan Idzikowski).
- The LDAPUserFolder will now use the LDAP database connection on a more persistent basis. A connection gets stored away and reused until it breaks or until the object is ghosted by Zope. This should speed up LDAP accesses in many situations.

2.4.64 1.4beta1 (2002-05-06)

- The unicode verification in utils.py could not deal with non-ascii characters (Tracker issue 122, thanks to Jan Idzikowski).
- The setting for encryption scheme was never set with the selection from the add form (thanks Dirk Datzert).
- The redirection after calling the constructor method has been changed to use RESPONSE.redirect to get over the idiosyncrasies of returning something from a self that is a factory dispatcher and not a container. (thanks to Dirk Datzert)
- Internal cache is no longer case-sensitive (suggested by Dirk Datzert)
- The list of available LDAP groups to map to Zope roles on the Groups tab is now sorted alphabetically (suggestion from Dirk Datzert).

2.4.65 1.3 (2002-04-16)

- Added workaround for the missing "crypt" module problem that only appears on a so-called OS from Redmond. (Tracker issues 119 and 120)
- Simplified package structure somewhat by adding a utils module that defines methods or constants used in the other modules.
- As a step to better unicode handling the LDAPUser now stores all attribute strings as unicode strings.

2.4.66 1.3beta1 (2002-04-05)

- Small bug in user object caching code that could lead to duplicate user objects being cached which only differ in capitalization.
- Added a couple small improvements and fixes as suggested by Dieter Maurer.
- Added workaround for the buggy windoze ldap.pyd that does not have a meaningful __version__ string (Tracker issue 118).
- Added the ability to map a LDAP group name to a Zope role name. In a nutshell, if LDAP group "Employee" is mapped to Zope role "Member" then anyone who authenticates through LDAP and is in LDAP group "Employee" will have "Employee" and "Member" in the list of roles for the user.

2.4.67 1.2 (2002-02-25)

- Dirk Datzert sent me a module that can encrypt strings using the SSHA encryption scheme. This prompted me to make user password encryption schemes selectable in the LDAPUserFolder configuration. Thanks Dirk!
- Added a new configuration toggle to set the use of SSL for the LDAP server connection. This option will be ignored and appear greyed-out on the Configure tab if the python-ldap module version is lower than 2.0.

- The detailed user data view on the Users tab will now show the full DN for every group that is listed as possible roles for a user. This will help everyone who has roles with the same name defined several times underneath their groups search base (Tracker item 107, thanks go to David Rideau).

2.4.68 1.2beta3 (2002-01-30)

- Not only did the importing semantics for the latest python-ldap modules change, some fundamental method signatures changed as well. This fixes the call to ldap.open which led to uncontrollable debugging output on the command line since 1.2beta2 (Tracker issue 106 - thanks to David Rideau).

2.4.69 1.2beta2 (2002-01-29)

- Ensure compatibility with the latest python-ldap releases which introduced an underdocumented and spurious change that mandates a different way of importing the ldap module. (Tracker issues 102 and 105) The recommended python-ldap module to use remains 1.10alpha3 which seems to work more reliably and has less bugs.

2.4.70 1.2beta1 (2002-01-28)

- If the authentication information was incorrect, the _searchResults method would raise the Unauthorized exception, which would make basic HTTP authentication boxes pop up even in cookie mode. It now returns an empty string and users will get the login page when in cookie mode. Tracker issue 87 - thanks to Eric Brun.
- Some browsers would not display the LDAP server port in the configuration tab due to a line break in the DTML for the form. The line break has been removed. (Tracker issue 103, thanks once again to Florent Guillaume)
- A little more extended debug logging in the _lookupuser method (suggested by Marc-Aurele Darche of IDEALX, Tracker issue 101)
- The RDN Attribute dropdown in the Configure tab now lists all attributes defined on the LDAP Schema tab. This allows the administrator to select any attribute if the LDAP setup violates RFC 2377. Suggestion and patch (thanks!) from Marc-Aurele Darche of IDEALX, Tracker issue 101.

2.4.71 1.1 (2001-12-14)

- Exclusively use direct string method invocation as opposed to using the string module
- Mistakenly cached the superuser account upon successful authentication. This would break the "Caches" tab because it tries to call a method on each cached user object that is not supported by the superuser API.
- A spurious "/" in the "Add Object" widget of the "Contents" tab broke rendering of the dropdown list in some browsers.
- Updated all help screens, some were outdated
- Updated the API help files

2.4.72 1.1beta3 (2001-12-07)

- Typo inside the method that adds user records led to an obscure error message and user roles would not be set (Tracker issue 95, submitted by M. A. Darche if IDEALX)
- The targets of some MessageDialog error screens for the management interface pointed to the wrong method name and caused a traceback.

2.4.73 1.1beta2 (2001-12-04)

- The administrator can now choose between reading group information from the LDAP server or storing it inside the LDAP User Folder itself. This feature has been added in response to Tracker issue 94. Thanks go to Colin Smith for bringing this configuration option to my attention.
- The Users tab in the ZMI would show tracebacks if the connection to the LDAP server produced an error. Now the real error is shown (Tracker Issue 93)
- Some details of the ZBabel transition did not work correctly, in particular the add screen blew up.

2.4.74 1.1beta1 (2001-12-01)

- ZBabel support, thanks go to Dirk Datzert.

2.4.75 1.0 (2001-11-18)

- Failed authentication in LDAP during a search operation will not simply re-raise the ldap.INVALID_CREDENTIALS anymore, but raise "Unauthorized", which will cause the browser to pop up an authentication dialog (Tracker issue 82 by Igor Stroh).
- Adding of users was broken because of a faulty invocation for translating record attributes into UTF-8 (Tracker issue 83, thanks go to Magnus Heino)
- Changing some attribute names led to the main management screen bombing out (Tracker Issue 84, thanks go to Magnus Heino)
- The way in which caches were used in getUser was broken since the last few betas and cache records with a fake passwords were sometimes used in a real validation context.
- LDAP User objects now have a __getattr__ that will look into the internal properties dictionary, meaning direct attribute access is now possible without the need to use getProperty(). The "Public User Attribute" machinery has been renamed to explain its new meaning in this context: It is used to map an attribute name from LDAP to another name that will also appear on the user object.

2.4.76 1.0 beta5 (unknown)

- Default roles were no longer applied (Tracker issue 73, thanks go to Eric Brun)
- Passwords can now contain colon (:) characters. This would break reading the authentication cookie in cookie mode before (Tracker issue 74, thanks to Eric Brun)
- Added logging to cache lookup successes and decided to simplify getUserById at the same time (Tracker issue 75, thanks to Eric Brun)
- The user ID as seen by zope is now guaranteed to be the same every time a user logs in, regardless of name capitalization (which LDAP ignores upon searching). Tracker issue 77, thanks go to Eric Brun.

2.4.77 1.0 beta4 (2001-11-08)

- Some missing imports related to the Cookie authentication and a tweak to the add screen so that the authentication mechanism choice gets applied correctly.

2.4.78 1.0 beta3 (2001-11-07)

- Added compatibility with WebDAV and FTP when in cookie mode
- Basic Auth is used as "last resort" when authenticating a user, meaning if you change to cookie mode from basic auth mode there will be no login screen if you were logged in.
- An attempt is made, while in cookie mode, to hand off authentication to the next user folder above if a user cannot be identified but might be valid in the user folder above.
- The life span of the authentication cookie (if cookie mode is in use) can now be set by the administrator.
- Vast amounts of code were removed by integrating the validate/authenticate/identify machinery better into the machinery provided by the BasicUserFolder base class.

2.4.79 1.0 beta2 (2001-10-15)

WARNING: As of this point the product is not API-compatible with the LDAPUserManager or LDAPLoginAdapter anymore. The API will change even more before the final 1.0-release.

Please experiment but don't use in production unless you really know what you are doing.

- The LDAP User Folder is much more flexible with the groups it reads and writes. Administrators can select the type of group upon creation and any code that manipulates group membership will do the "right thing". Any code retrieving group information can work with varying kinds of groups now. This should ease integration with M$ Active Directory.
- The user id shown in the "Cache" tab is now a link that will bring up the user data for the specific user.
- The "Configuration" tab has been refactored to be more visually pleasing and less space consuming.
- The "LDAP Schema" tab has been completely revamped. The fields for inputting a mapped public name that will show up on the user object to an LDAP attribute has been integrated into the main display and can be triggered by adding attributes with the optional "Public name" value filled in.
- The "log verbosity" setting has been moved onto the "Log" tab.
- The horribly kludgy "getGroupsWithInconsistentRecords" method, introduced in LDAPLoginAdapter 1.3beta4, was dropped.
- Any attempt to be backwards-compatible with Python 1.5.x was removed. This product needs Zope 2.4.x, which in turn depends on Python 2.1.x or higher.
- The "Advanced" configuration tab has been removed after consolidating its functionality into other screens.

2.4.80 1.0 beta1 (2001-10-09)

This product combines the LDAPLoginAdapter and LDAPUserManager products into a single package. It is designed to supplant both of them and further development, apart from urgent bug fixes, will be limited to this product.

Important Note : This product is compatible with Zope versions 2.4 and up, which necessitates Python 2.0 and up. Older Zope sites should continue to use the LDAPLoginAdapter/LDAPUserManager combo.

This first version is simply a combination of all contents from all classes used in the LDAPLoginAdapter and LDAPUserManager code. Future plans (probably version 2.0 and up) will include a complete refactoring with class separation of

- Storage backend operations
- User Folder API and setting of properties on the LDAPUserFolder

Other cleanups in this first cut involve the following:

- throwing out of code that tried to compensate for cases where a user record's DN had spacing different from its reference inside a group record
- throwing out of code that tried to fake unicode support for Python versions prior to 1.6
- Some refactoring of ZMI screens, there will undoubtedly be more in the future.

CHAPTER 3

Support

If you need commercial support for this software package, please see https://www.zetwork.com.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search
- glossary