
Zope 2 Documentation

Release 2.12

Zope Developers Community

December 10, 2013

Contents

Note: The Zope 2.12.x release series is no longer maintained and won't see security fixes either anymore. Support ended in October 2013.

Contents:

What's new in Zope 2.12

The article explains the new high-level features and changes found in this version of Zope 2.

You can have a look at the detailed change log to learn about all minor new features and bugs being solved in this release.

1.1 Support for newer Python versions

Zope 2 has supported and required Python 2.4 since its 2.9 release in summer 2006. Later versions of Python have so far been unsupported by Zope 2.

This version of Zope 2 adds support for Python 2.6 instead. As neither Python 2.4 nor Python 2.5 are officially maintained any longer. Python 2.4 on 64-bit platforms is known to be incompatible with Zope 2.12. On 32-bit it could work at your own risk. While there are no known incompatibilities with Python 2.5 it is not recommended.

Python 3 is a backwards incompatible release of Python and not supported. At this point there is no concrete roadmap for adoption of Python 3. It is expected to be a question of multiple major Zope 2 releases or years, though.

1.2 Fully eggified

Zope 2 itself is now fully eggified and compatible with `setuptools`. You can use popular tools like `easy_install` or `zc.buildout` to install Zope 2.

Releases of Zope 2 can be found at and will be installable from the Python package index at <http://pypi.python.org/pypi/Zope2>.

The repackaging of Zope 2 into an eggified form and accompanying changes to the file system layout have caused a number of changes. The environment variables `SOFTWARE_HOME` and `ZOPE_HOME` are no longer available nor set in the control scripts. If you need to access data files inside the Zope 2 package, you can for example use `import os, OFS; os.path.dirname(OFS.__file__)` to locate the files inside the OFS package.

In general it is discouraged to rely on the `lib/python` and `Products` directories to make code available to the running Zope process. While these mechanisms continue to work, you are encouraged to use normal distutils or setuptools managed packages and add these to your `sys.path` using any of the standard Python mechanisms. To create isolated Python environments both `zc.buildout` and `virtualenv` are in wide-spread use.

1.3 Zope 3

This version of Zope 2 does no longer includes a full version of the Zope 3 application server nor a specific release of it. Instead it includes a minimal set of packages from the former Zope 3 project. Only those packages actually used by Zope 2 itself are still included. Examples of such packages include `zope.interface`, `zope.component`, `zope.i18n`, `zope.tal` and many more.

Zope 2.12 marks a transitional release, where a number of `zope.app` packages are still included. An ongoing effort to refactor those packages into more reusable and minimal packages is going on and will result in no `zope.app` packages being used or shipped with anymore as part of Zope 2. Zope 2.13 does no longer ship with any `zope.app` packages.

If you are using `zope.app` packages inside your own codebase, you should make sure to declare these as dependencies of your packages or otherwise follow the refactoring effort and remove your dependency on these packages.

As a result of this refocussing on more minimal dependencies, the number of packages included by Zope 2 could be dramatically reduced from about 120 additional packages to just over 70. The total code size of Zope 2 and its dependencies has decreased by over 200,000 lines of code as a result.

1.4 ZODB 3.9

This version of Zope 2 includes the latest version of the **ZODB (3.9)**. It has a multitude of new configuration options and bug fixes. File storages have gotten native support for blob storages and demo storages have been expanded extensively. There is a large number of options to tune ZEO servers and clients in large scale environments and control cache invalidation and packaging to a much wider degree.

You can read more about the detailed changes in the [ZODB3 change log](#) for version 3.9.

1.5 Module cleanup

As with every release of Zope 2 this version has removed various modules which have been deprecated in prior versions.

Most notably `ZClasses` and supporting modules have been removed entirely from Zope 2. As a result the persistent product registry has been made optional, but is still enabled by default. If your application does not rely on the registry, you can now disable it by specifying:

```
enable-product-installation off
```

inside your `zope.conf` file. With the option turned off Zope will no longer write any new transactions to your database during startup in most cases.

With the upgrade to ZODB 3.9 database-level version support is no longer available. Many of the modules in *Products.OFSP* have been removed as a result. The low level API to load objects from the database has lost its version argument as a result of this.

1.6 Documentation updates

Zope 2 now uses [Sphinx](#) to create pleasant HTML representations of its documentation. An effort is underway to update the publicly available documentation using Sphinx at <http://docs.zope.org/>.

So far the Zope 2 Book, the Zope Developers Guide and many smaller articles have been converted to reStructuredText and their content updated.

1.7 Acquisition redux

The short technical version of this change is: “Acquisition has been made aware of `__parent__` pointers”. What sounds like a small change is actually a major step in the integration story for Zope components based technology into Zope 2.

While integrating the Zope component architecture and its many concepts into Zope 2 an integration layer called Five (Zope 2 + 3) has been created. One of the major reasons for the necessity of an integration layer has been in the way Zope 2 was tightly coupled to the concept of Acquisition. The entire security machinery, object traversal and publication has been relying on this.

All classes, which wanted to interact with Zope 2 in any non-trivial way, had to inherit from the Acquisition base classes. As a result almost no external package could directly work inside Zope 2 but required an integration layer.

With this version of Zope 2, objects have a new option of providing location awareness to Zope APIs. This new option is to provide an explicit parent pointer in the `__parent__` attribute, much like specified by the `ILocation` API from `zope.location`. Browser views and other location-dependent components implement `ILocation` already.

Classes adhering to this convention need to get `__parent__` pointers set to their container object, when being put into the container. Code that operates on such objects can then walk up the containment hierarchy by following the pointers. In Acquisition based classes no information would be stored on the objects, but Acquisition wrappers are constructed around the objects instead. Only those wrappers would hold the container references. The Acquisition wrapping relies on the objects to provide an `__of__` method as done by the Acquisition base classes.

The most common way of getting the container of an instance is to call:

```
from Acquisition import aq_parent
```

```
container = aq_parent(instance)
```

For instances providing the `ILocation` interface the common way is:

```
container = instance.__parent__
```

There are various `aq_*` methods available for various other tasks related to locating objects in the containment hierarchy. So far virtually all objects in Zope 2 would participate in Acquisition. As a side-effect many people relied on Acquisition wrappers to be found around their objects. This caused code to rely on accessing the `aq_*` methods as attributes of the wrapper:

```
container = instance.aq_parent
```

While all the existing API's still work as before, Acquisition now respects `__parent__` pointers to find the container for an object. It will also not unconditionally try to call the `__of__` method of objects anymore, but protect it with a proper interface check:

```
from Acquisition.interfaces import IAcquirer
```

```
if IAcquirer.providedBy(instance):
    instance = instance.__of__(container)
```

In addition to this check you should no longer rely on the `aq_*` methods to be available as attributes. While all code inside Zope 2 itself still supports this, it does no longer rely on those but makes proper use of the functions provided by the Acquisition package.

To understand the interaction between the new and old approach here is a little example:

```
>>> class Location(object):
...     def __init__(self, name):
...         self.__name__ = name
...     def __repr__(self):
```

```
...         return self.__name__

# Create an Acquisition variant of the class:

>>> import Acquisition
>>> class Implicit(Location, Acquisition.Implicit):
...     pass

# Create two implicit instances:

>>> root = Implicit('root')
>>> folder = Implicit('folder')

# And two new Acquisition-free instances:

>>> container = Location('container')
>>> item = Location('item')

# Provide the containment hints:

>>> folder = folder.__of__(root)
>>> container.__parent__ = folder
>>> item.__parent__ = container

# Test the containment chain:

>>> from Acquisition import aq_parent
>>> aq_parent(container)
folder

>>> from Acquisition import aq_chain
>>> aq_chain(item)
[item, container, folder, root]

# Explicit pointers take precedence over Acquisition wrappers:

>>> item2 = Implicit('item2')
>>> item2 = item2.__of__(folder)
>>> item2.__parent__ = container

>>> aq_chain(item2)
[item2, container, folder, root]
```

For a less abstract example, you so far had to do:

```
>>> from Acquisition import aq_inner
>>> from Acquisition import aq_parent
>>> from Products import Five

>>> class MyView(Five.browser.BrowserView):
...
...     def do_something(self):
...         container = aq_parent(aq_inner(self.context))
```

Instead you can now do:

```
>>> import zope.publisher.browser

>>> class MyView(zope.publisher.browser.BrowserView):
```

```

...
...     def do_something(self):
...         container = aq_parent(self.context)

```

As the zope.publisher BrowserView supports the ILocation interface, all of this works automatically. A view considers its context as its parent as before, but no longer needs Acquisition wrapping for the Acquisition machinery to understand this. The next time you want to use a package or make your own code more reusable outside of Zope 2, this should be of tremendous help.

1.8 Object managers and IContainer

One of the fundamental parts of Zope 2 is the object file system as implemented in the *OFS* package. A central part of this package is an underlying class called *ObjectManager*. It is a base class of the standard *Folder* used for many container-ish classes inside Zope 2.

The API to access objects in an object manager or to add objects to one has been written many years ago. Since those times Python itself has gotten standard ways to access objects in containers and work with them. Those Python API's are most familiar to most developers working with Zope. The Zope components libraries have formalized those API's into the general IContainer interface in the zope.container package. In this version of Zope 2 the standard OFS ObjectManager fully implements this IContainer interface in addition to its old API.

```

>>> from zope.container.interfaces import IContainer
>>> from OFS.ObjectManager import ObjectManager
>>> IContainer.implementedBy(ObjectManager)
True

```

You can now write your code in a way that no longer ties it to object managers alone, but can support any class implementing IContainer instead. In conjunction with the Acquisition changes above, this will increase your chances of being able to reuse existing packages not specifically written for Zope 2 in a major way.

Here's an example of how you did work with object managers before:

```

>>> from OFS.Folder import Folder
>>> from OFS.SimpleItem import SimpleItem

>>> folder = Folder('folder')
>>> item1 = SimpleItem('item1')
>>> item2 = SimpleItem('item2')

>>> result = folder._setObject('item1', item1)
>>> result = folder._setObject('item2', item2)

>>> folder.objectIds()
['item1', 'item2']

>>> folder.objectValues()
[<SimpleItem at folder/>, <SimpleItem at folder/>]

>>> if folder.hasObject('item2')
...     folder._delObject('item2')

```

Instead of this special API, you can now use:

```

>>> from OFS.Folder import Folder
>>> from OFS.SimpleItem import SimpleItem

>>> folder = Folder('folder')

```

```
>>> item1 = SimpleItem('item1')
>>> item2 = SimpleItem('item2')

>>> folder['item1'] = item1
>>> folder['item2'] = item2

>>> folder.keys()
['item1', 'item2']

>>> folder.values()
[<SimpleItem at folder/>, <SimpleItem at folder/>]

>>> folder.get('item1')
<SimpleItem at folder/>

>>> if 'item2' in folder:
...     del folder['item2']

>>> folder.items()
[('item1', <SimpleItem at folder/>)]
```

Installing and Zope with `zc.buildout`

This document describes how to get going with Zope using `zc.buildout`.

2.1 About `zc.buildout`

`zc.buildout` is a powerful tool for creating repeatable builds of a given software configuration and environment. The Zope developers use `zc.buildout` to develop Zope itself, as well as the underlying packages it uses.

2.2 Prerequisites

In order to use Zope, you must have the following pre-requisites available:

- A supported version of Python, including the development support if installed from system-level packages. Supported versions include:
 - 2.6.x
- Zope needs the Python `zlib` module to be importable. If you are building your own Python from source, please be sure that you have the headers installed which correspond to your system's `zlib`.
- A C compiler capable of building extension modules for your Python (`gcc` recommended). This is not necessary for Windows as binary releases of the parts that would need compiling are always made available.
- If you wish to install Zope as a Service on Windows, you will need to have the `pywin32` package installed.

2.3 Installing standalone Zope using `zc.buildout`

In this configuration, we use `zc.buildout` to install the Zope software, but then generate server “instances” outside the buildout environment.

2.3.1 Installing the Zope software

Installing the Zope software using `zc.buildout` involves the following steps:

- Download the Zope 2 source distribution from [PyPI](#)
- Bootstrap the buildout
- Run the buildout

On Linux, this can be done as follows:

```
$ wget http://pypi.python.org/packages/source/Z/Zope2/Zope2-<Zope version>.tar.gz
$ tar xfvz Zope2-<Zope version>.tar.gz
$ cd Zope2-<Zope version>
$ /path/to/your/python bootstrap/bootstrap.py
$ bin/buildout
```

2.3.2 Creating a Zope instance

Once you’ve installed Zope, you will need to create an “instance home”. This is a directory that contains configuration and data for a Zope server process. The instance home is created using the `mkzopeinstance` script:

```
$ bin/mkzopeinstance
```

You can specify the Python interpreter to use for the instance explicitly:

```
$ bin/mkzopeinstance --python=$PWD/bin/zopecy
```

You will be asked to provide a user name and password for an administrator’s account during `mkzopeinstance`. To see the available command-line options, run the script with the `--help` option:

```
$ bin/mkzopeinstance --help
```

Note: The traditional “inplace” build is no longer supported. If using `mkzopeinstance`, always do so outside the buildout environment.

2.4 Creating a buildout-based Zope instance

Rather than installing Zope separately from your instance, you may wish to use `zc.buildout` to create a self-contained environment, containing both the Zope software and the configuration and data for your server. This procedure involves the following steps:

- Create the home directory for the buildout, including `etc`, `log` and `var` subdirectories.
- Fetch the buildout bootstrap script into the environment.
- Fetch the version file into the environment, for example: <https://raw.githubusercontent.com/zopefoundation/Zope/2.12.28/versions.cfg>
- Create a buildout configuration as follows:

buildout.cfg

```
[buildout]
parts = instance
extends = versions.cfg

[instance]
recipe = zc.recipe.egg
eggs = Zope2
interpreter = py
scripts = runzope zopectl
initialization =
    import sys
    sys.argv[1:1] = ['-C', r'${buildout:directory}/etc/zope.conf']
```

This is the minimum but all the usual buildout techniques can be used.

- Bootstrap the buildout
- Run the buildout
- Create a Zope configuration file. A minimal version would be:

etc/zope.cfg

```
%define INSTANCE <path to your instance directory>

python $INSTANCE/bin/py[.exe on Windows]

instancehome $INSTANCE
```

A fully-annotated sample can be found in the Zope2 egg:

```
$ cat eggs/Zope2--*/Zope2/utilities/skel/etc/zope.conf.in
```

<rest of the stuff that goes into a zope.conf, e.g. databases and log files.>

An example session:

```
$ mkdir /path/to/instance
$ cd /path/to/instance
$ mkdir etc logs var
$ wget http://svn.zope.org/zc.buildout/trunk/bootstrap/bootstrap.py
$ vi buildout.cfg
$ /path/to/your/python bootstrap.py
$ bin/buildout
$ cat eggs/Zope2--*/Zope2/utilities/skel/etc/zope.conf.in > etc/zope.conf
$ vi etc/zope.conf # replace <<INSTANCE_HOME>> with buildout directory
$ bin/zopectl start
```

In the bin subdirectory of your instance directory, you will find runzope and zopectl scripts that can be used as normal.

You can use zopectl interactively as a command shell by just calling it without any arguments. Try help there and help <command> to find out about additionally commands of zopectl. These commands also work at the command line.

Note that there are there are recipes such as [plone.recipe.zope2instance](#) which can be used to automate this whole process.

After installation, refer to *Configuring and Running Zope* for documentation on configuring and running Zope.

Configuring and Running Zope

Whichever method you used to install Zope and create a server instance (see *Installing and Zope with `zc.buildout`* and *INSTALL-buildout*), the end result is configured and operated the same way.

3.1 Configuring Zope

Your instance's configuration is defined in its `etc/zope.conf` file. Unless you created the file manually, that file should contain fully- annotated examples of each directive.

You can also pass an explicit configuration file on the commandline:

```
$ /path/to/zope/instance/bin/zopectl -c /tmp/other.conf show
...
Config file: /tmp/other.conf
```

When starting Zope, if you see errors indicating that an address is in use, then you may have to change the ports Zope uses for HTTP or FTP. The default HTTP and FTP ports used by Zope are 8080 and 8021 respectively. You can change the ports used by editing `.etc/zope.conf` appropriately.

The section in the configuration file looks like this:

```
<http-server>
  # valid keys are "address" and "force-connection-close"
  address 8080
  # force-connection-close on
</http-server>
```

The address can just be a port number as shown, or a host:port pair to bind only to a specific interface.

After making any changes to the configuration file, you need to restart any running Zope server for the affected instance before changes are in effect.

3.2 Running Zope in the Foreground

To run Zope without detaching from the console, use the `fg` command (short for foreground):

```
$ /path/to/zope/instance/bin/zopectl fg
```

In this mode, Zope emits its log messages to the console, and does not detach from the terminal.

3.3 Running Zope as a Daemon

Once an instance home has been created, the Zope server can now be started using this command:

```
$ /path/to/zope/instance/bin/zopectl start
```

During startup, Zope emits log messages into */path/to/zope/instance/log/event.log*. You can examine it with the usual tools (*cat*, *more*, *tail*, etc) and see if there are any errors preventing Zope from starting.

Note: For this to work on Windows, the Zope instance must be installed as a Service. This is done with:

```
bin\zopectl install
```

If you later want to remove this Service, do the following:

```
bin\zopectl remove
```

For the full list of options available for setting up Zope as a Windows Service, do:

```
bin\zopectl install --help
```

3.4 Integrating with System Startup

zopectl can be linked as rc-script in the usual start directories on linux or other System V unix variants.

You can use *zopectl* interactively as a command shell by just calling it without any arguments. Try *help* there and *help <command>* to find out about additional commands of *zopectl*. These commands also work at the command line.

Note: On Windows, a Service can be installed and set to start automatically with the following:

```
bin\zopectl install --startup=auto
```

3.5 Logging In To Zope

Once you've started Zope, you can then connect to the Zope webserver by directing your browser to:

```
http://yourhost:8080/manage
```

where 'yourhost' is the DNS name or IP address of the machine running Zope. If you changed the HTTP port as described, use the port you configured.

You will be prompted for a user name and password. Use the user name and password you provided in response to the prompts issued during the "make instance" process.

Now you're off and running! You should be looking at the Zope management screen which is divided into two frames. On the left you can navigate between Zope objects and on the right you can edit them by selecting different management functions with the tabs at the top of the frame.

If you haven't used Zope before, you should head to the Zope web site and read some documentation. The Zope Documentation section is a good place to start. You can access it at <http://docs.zope.org/>

3.6 Troubleshooting

- This version of Zope requires Python 2.6.4 or better. It will *not* run with Python 3.x.
- The Python you run Zope with *must* have threads compiled in, which is the case for a vanilla build. Warning: Zope will not run with a Python version that uses `libpth`. You *must* use `libpthread`.
- To build Python extensions you need to have Python configuration information available. If your Python comes from an RPM you may need the `python-devel` (or `python-dev`) package installed too. If you built Python from source all the configuration information should already be available.
- See the *Changelog* for important notes on this version of Zope.

3.7 Adding extra commands to Zope

It is possible to add extra commands to `zopectl` by defining *entry points* in `setup.py`. Commands have to be put in the `zopectl.command` group:

```
setup(name="MyPackage",
      ....
      entry_points="""
      [zopectl.command]
      init_app = mypackage.commands:init_application
      """)
```

Note: Due to an implementation detail of `zopectl` you can not use a minus character (-) in the command name.

This adds a `init_app` command that can be used directly from the command line:

```
bin\zopectl init_app
```

The command must be implemented as a Python callable. It will be called with two parameters: the Zope2 application and a list with all command line arguments. Here is a basic example:

```
def init_application(app, args):
    print 'Initializing the application'
```

Make sure the callable can be imported without side-effects, such as setting up the database connection used by Zope 2.

Special Users

Because Zope is managed through the web, user names and passwords must be used to assure that only authorized people can make changes to a Zope installation.

4.1 Adding Managers

If you need to add a Manager to an existing Zope instance, you can do this using *zopectl* as follows:

```
zopectl adduser 'name' 'password'
```

4.2 The Initial User

An initial username and password is needed to “bootstrap” the creation of normal managers of your Zope site. This is accomplished through the use of the ‘inituser’ file in the directory specified as the instance home.

The first time Zope starts, it will detect that no users have been defined in the root user folder. It will search for the ‘inituser’ file and, if it exists, will add the user defined in the file to the root user folder.

Normally, ‘inituser’ is created by the Zope install scripts. Either the installer prompts for the password or a randomly generated password is created and displayed at the end of the build script.

You can use the ‘zpasswd.py’ script to create ‘inituser’ yourself. Execute ‘zpasswd.py’ like this:

```
python zpasswd.py inituser
```

The script will prompt you for the name, password, and allowed domains. The default is to encode the password with SHA, so please remember this password as there is no way to recover it (although ‘zpasswd.py’ lets you reset it.)

4.3 The Emergency User

In some situations you may need to bypass normal security controls because you have lost your password or because the security settings have been mixed up. Zope provides a facility called an “emergency user” so that you can reset passwords and correct security settings.

The emergency user password must be defined outside the application user interface. It is defined in the 'access' file located in the Zope directory. It should be readable only by the user as which your web server runs.

To create the emergency user, use 'zpasswd.py' to create the 'access' file like this:

```
python zpasswd.py access
```

In order to provide a somewhat higher level of security, various encoding schemes are supported which provide access to either SHA-1 encryption or the standard UNIX crypt facility if it has been compiled into Python. Unless you have some special requirements (see below), you should use the SHA-1 facility, which is the default.

4.4 Format of 'inituser' and 'access'

A password file should consist of a single line of the form:

```
name:password
```

Note that you may also add an optional third component to the line in the access file to restrict access by domain. For example, the line:

```
mario:nintendoRules:*.mydomain.com
```

in your 'access' file will only allow permit emergency user access from **.mydomain.com* machines. Attempts to access the system from other domains will fail, even if the correct emergency user name and password are used.

Please note that if you use the ZServer monitor capability, you will need to run with a clear text password.

Filesystem Permissions

You need to set permissions on the directory Zope uses to store its data. This will normally be the *var* directory in the instance home. Zope needs to read and write data to this directory. Before running Zope you should ensure that you give adequate permissions to this directory for the userid Zope will run under.

Depending on how you choose to run Zope you will need to give different permissions to the directory. If you use Zope with an existing web server, it will probably run Zope as 'nobody'. In this case 'nobody' needs read and write permissions to the var directory.

If you change the way you run Zope, you may need to modify the permissions of the directory and the files in it to allow Zope to read and write under its changed userid.

Zope effective user support

Note: It is best practice to run Zope behind a reverse proxy like Apache, Squid or Varnish. In this case, you do not need to run or install Zope with root privileges, since the reverse proxy will bind to port 80 and proxy back all request to Zope running on an unprivileged port.

Zope can bind its network service to low ports such as 21 (FTP) and 80 (HTTP). In order to bind to low ports, Zope must be started as the root user. However, Zope will only run as root long enough to bind to these low ports. It will then attempt to setuid to a less privileged user.

You must specify the user to which Zope will attempt to setuid by changing the ‘effective-user’ parameter in the `zope.conf` configuration file to an existing username or UID. All runtime files will be written as this user. If you do not specify an ‘effective-user’ in the configuration file, and you attempt to start Zope, it will refuse to start.

Zope additionally emits a warning if you specify ‘nobody’ as the ‘effective-user’. The rationale for this warning stems from the fact that, historically, many other UNIX services dropped privileges to the ‘nobody’ account after starting as root. Any security defects in these services could cause someone to gain access as the ‘nobody’ account on your system. If someone was to gain control of your ‘nobody’ account they could compromise your Zope files.

The most important thing to remember about effective user support is that you don’t have to start Zope as root unless you want to listen for requests on low ports (ports beneath 1024). In fact, if you don’t have this need, you are much better off just starting Zope under a dedicated user account.

Signals (POSIX only)

Signals are a POSIX inter-process communications mechanism. If you are using Windows then this documentation does not apply.

Zope responds to signals which are sent to the process id specified in the file '\$INSTANCE_HOME/var/Z2.pid':

SIGHUP - close open database connections, then restart the server process. A idiom for restarting a Zope server is:

```
kill -HUP `cat $INSTANCE_HOME/var/z2.pid`
```

SIGTERM - close open database connections then shut down. A common idiom for shutting down Zope is:

```
kill -TERM `cat $INSTANCE_HOME/var/Z2.pid`
```

SIGINT - same as SIGTERM

SIGUSR2 - close and re-open all Zope log files (z2.log, event log, detailed log.) A common idiom after rotating Zope log files is:

```
kill -USR2 `cat $INSTANCE_HOME/var/z2.pid`
```

Running Zope in Debug Mode

If you wish to run Zope in debug mode, set the ‘debug-mode’ configuration file parameter to ‘on’ (this is the default). This will have the following effects:

- On UNIX, Zope will not detach from the controlling terminal.
- The `Z_DEBUG_MODE` environment variable gets set, which causes behavioral changes to Zope appropriate for software development. See the configuration file description of ‘debug-mode’ for more information.

8.1 Using ‘zopectl debug’

A utility known as ‘zopectl’ is installed into generated instance homes. You can use it to inspect a Zope instance’s running state via an interactive Python interpreter by passing zopectl the ‘debug’ parameter on the command line. The ‘top-level’ Zope object (the root folder) will be bound to the name ‘app’ within the interpreter. You can then use normal Python method calls against app and use the Python interpreter normally to inspect results:

```
[chrism@james Trunk]$ bin/zopectl debug
Starting debugger (the name "app" is bound to the top-level Zope object)
>>> app.objectIds()
['acl_users', 'Control_Panel', 'temp_folder', 'browser_id_manager', 'session_data_manager', 'error_lo
>>>
```

Changelog

This file contains change information for the current Zope release. Change information for previous versions of Zope can be found at <http://docs.zope.org/zope2/>

9.1 2.12.28 (2013-07-16)

- LP #1095343: Prevent sandbox escape via `BaseRequest.traverseName`.
- LP #1094144: Prevent arbitrary redirections via faked “CANCEL” buttons.
- LP #1094221: Add permissions to some unprotected methods of `OFS.ObjectManager`
- LP #1094049: Prevent zlib-based DoS when parsing the cookie containing paste tokens.
- Updated distributions:
 - `DateTime` = 2.12.7

9.2 2.12.27 (2013-05-01)

- LP #1114688: Defend against minidom-based DoS in webdav. (Patch from Christian Heimes).
- LP #978980: Protect views of ZPT source with ‘View Management Screens’ permission.

9.3 2.12.26 (2012-10-31)

- LP #1071067: Use a stronger random number generator and a constant time comparison function.
- LP #930812: Scrub headers a bit more.

9.4 2.12.25 (2012-09-18)

- Explicitly declared ZTUtils APIs as public (repairs breakages in apps following fix for LP #1047318).

9.5 2.12.24 (2012-09-09)

- LP #1047318: Tighten import restrictions for restricted code.
- Fix a bug in `ZopeSecurityPolicy.py`. Global variable `rolesForPermissionOn` could be overridden if `__role__` had custom `rolesForPermissionOn`.
- Updated distributions:
 - `zdaemon` = 2.0.7
 - `zope.tales` = 3.4.1

9.6 2.12.23 (2012-02-20)

- Note end-of-life timeline: Zope 2.12.x is now in security-fix-only mode and will continue to see security updates until October 2013, the same as Python 2.6.x does.
- Ensure that `ObjectManager`'s `get` and `__getitem__` methods return only “items” (no attributes / methods from the class or from acquisition). Thanks to Richard Mitchell at Netsight for the report.
- Updated distributions:
 - `distribute` = 0.6.24
 - `Products.ZSQLMethods` = 2.13.4
 - `zope.catalog` = 3.8.2
 - `zope.componentvocabulary` = 1.0.1
 - `zope.datetime` = 3.4.1
 - `zope.deprecation` = 3.4.1
 - `zope.documenttemplate` = 3.4.3
 - `zope.index` = 3.6.3
 - `zope.keyreference` = 3.6.4
 - `zope.mkzeoinstance` = 3.9.5
 - `zope.session` = 3.9.5
 - `zope.testing` = 3.9.7

9.7 2.12.22 (2012-01-18)

- Prevent a cross-site-scripting attack against the default standard error message handling. (CVE-2010-1104).

9.8 2.12.21 (2011-12-12)

- Restrict the available functions in `DocumentTemplate.sequence` to public API's of `zope.sequencesort`.
- LP #1079238: Turn `UndoSupport.get_request_var_or_attr` helper into a private API.

- Provide security declarations for *BTreeFolder2Base* class, instead of only for *BTreeFolder2* via the *OFS.Folder* mix-in.
- LP #902068: Fixed missing security declaration for *ObjectManager* class.
- Fixed serious authentication vulnerability in stock configuration.

9.9 2.12.20 (2011-10-04)

- Fixed serious arbitrary code execution issue (CVE 2011-3587) <http://zope2.zope.org/news/security-vulnerability-announcement-cve-2011-3587>

9.10 2.12.19 (2011-06-28)

9.10.1 Bugs Fixed

- Fixed a serious privilege escalation issue. For more information see: <http://plone.org/products/plone/security/advisories/20110622>

9.10.2 Features

- Updated distributions:
 - Acquisition = 2.13.8

9.11 2.12.18 (2011-05-08)

9.11.1 Features

- Updated distributions:
 - zope.sendmail = 3.5.3

9.12 2.12.17 (2011-04-03)

9.12.1 Bugs Fixed

- Corrected copyright information shown in the ZMI.

9.12.2 Features

- Updated distributions:
 - Acquisition = 2.13.7
 - distribute = 0.6.15
 - docutils = 0.7

- python-gettext = 1.1.1
- pytz = 2011e
- transaction = 1.0.1

9.13 2.12.16 (2011-02-23)

9.13.1 Bugs Fixed

- LP #713820: Remove docstrings from various methods in ZCatalog, as they shouldn't be web-publishable.

9.13.2 Features

- Updated distributions:
 - Acquisition = 2.13.6

9.14 2.12.15 (2011-02-06)

9.14.1 Bugs Fixed

- LP #713253: Prevent publication of acquired attributes, where the acquired object does not have a docstring.
- Fix *LazyMap* to avoid unnecessary function calls.

9.15 2.12.14 (2010-12-07)

9.15.1 Bugs Fixed

- Fixed argument parsing for entrypoint based zopectl commands.
- Fixed the usage of `pstats.Stats()` output stream. The *Control_Panel/DebugInfo/manage_profile* ZMI view was broken in Python 2.5+.
- Use `cProfile` where possible for the *Control_Panel/DebugInfo/manage_profile* ZMI view.

9.16 2.12.13 (2010-11-06)

9.16.1 Bugs Fixed

- Fixed Chameleon compatibility in *ptEdit.zpt* of *Products.PageTemplates*.

9.16.2 Features Added

- LP #659968: Added support for level argument to the `__import__` function as introduced in Python 2.5. Currently only `level=-1` is supported.
- Updated distributions:
 - `DateTime` = 2.12.6

9.17 2.12.12 (2010-10-02)

9.17.1 Bugs Fixed

- Adjusted overflow logic in `DateIndex` and `DateRangeIndex` to work with latest ZODB 3.9.7.
- Fixed `testZODBCompat` tests in `ZopeTestCase` to match modern ZODB semantics.
- Fixed unit test that failed on fast Windows machines.
- Fixed `OverflowError` in `Products.ZCatalog.Lazy` on 64bit Python on Windows.
- LP #642728: Fixed `TypeError` on nested multi part messages in `MailHost`.

9.17.2 Features Added

- Add ability to define extra `zopectl` commands via `setuptools` entrypoints.
- Updated distributions:
 - `Acquisition` = 2.13.5
 - `distribute` = 0.6.14
 - `lxml` = 2.2.8
 - `z3c.checkversions` = 0.4.1
 - `zc.buildout` = 1.4.4
 - `zc.recipe.testrunner` = 1.2.1
 - `ZODB3` = 3.9.7
 - `zope.app.error` = 3.5.3
 - `zope.app.interface` = 3.5.2
 - `zope.app.locales` = 3.6.2
 - `zope.app.session` = 3.6.2
 - `zope.deferredimport` = 3.5.3
 - `zope.i18n` = 3.7.4
 - `zope.i18nmessageid` = 3.5.3
 - `zope.index` = 3.6.1
 - `zope.lifecycleevent` = 3.6.2
 - `zope.pagetemplate` = 3.5.2
 - `zope.principalregistry` = 3.7.1

- zope.proxy = 3.6.1
- zope.security = 3.7.4
- zope.sendmail = 3.5.2
- zope.session = 3.9.3

9.18 2.12.11 (2010-09-09)

9.18.1 Bugs Fixed

- BasicUserFolder and SiteErrorLog: Don't raise string exceptions.

9.18.2 Features Added

- LP #193122: New method getVirtualRoot added to the Request class.
- Added forward-compatibility shims for some frequently used modules moved in Zope 2.13.
- Updated packages:
 - Acquisition = 2.13.4
 - DateTime = 2.12.5
 - Products.ZSQLMethods = 2.13.3

9.19 2.12.10 (2010-07-20)

9.19.1 Bugs Fixed

- Fix support for non-public permission attributes in the browser:view directive so that attributes which are not included in allowed_interface or allowed_attributes but which have declarations from a base class's security info don't get their security overwritten to be private.
- LP #143755: Also catch TypeError when trying to determine an indexable value for an object in PluginIndexes.common.UnIndex
- LP #143533: Instead of showing "0.0.0.0" as the SERVER_NAME request variable when no specific listening IP is configured for the HTTP server, do a socket lookup to show the current server's fully qualified name.
- LP #143722: Added missing permission to ObjectManager.manage_hasId, which prevented renaming files and folders via FTP.
- LP #143564: Request.resolve_url did not correctly re-raise exceptions encountered during path traversal.
- LP #143273: Enable the dtml-var modifiers url_quote, url_unquote, url_quote_plus and url_unquote_plus to handle unicode strings.

9.19.2 Features Added

- Warn when App.ImageFile.ImageFile receives a relative path with no prefix, and then has to assume the path to be relative to "software home". This behaviour is deprecated as packages can be factored out to their own distribution, making the "software home" relative path meaningless.

9.20 2.12.9 (2010-07-13)

9.20.1 Bugs Fixed

- Avoid problems in removing cache managers introduced in Zope 2.12.8.
- LP #143531: Fix broken object so they give access to their state.
- LP #578326: Add support for non-public permission attributes in the browser:view directive.

9.20.2 Features Added

- Factored out the *Products.ZSQLMethods* into its own distribution. The distribution also includes the *Shared.DC.ZRDB* code. The code is no longer included in Zope 2.13. If you rely on it, please specify a dependency on the new *Products.ZSQLMethods* distribution. This will only work for code that requires at least Zope 2.12.9.
- Made both *Shared* and *Shared.DC* namespace packages.
- Updated packages:
 - DateTime = 2.12.4

9.21 2.12.8 (2010-06-25)

9.21.1 Bugs Fixed

- LP #597594: Performance optimization in OFS.subscriber.maybeWarnDeprecated.
- LP #143639: When the last cache manager in a container is deleted, we need to remove all traces of it from the container.
- LP #143619: Make sure to remove a RAMCache's contents when the ZODB object is removed.
- LP #143403: Prevent accidental acquisition of objectValues during recursive ownership changes when the changed object has no objectValues method.
- LP #374818: Use module-provided functions as opposed to the old "folder methods" when creating folders and user folders in ZopeTestCase.
- LP #143946: Provide a more informative error message when a WebDAV PUT fails.

9.21.2 Features Added

- Updated packages:
 - ExtensionClass = 2.13.2
 - Missing = 2.13.1
 - Persistence = 2.13.2
 - Sphinx = 0.6.7
- Added `setSortKey()` method to the `Shared.DC.ZRDB.TM.TM` class to allow database connections to specify the commit order without needing to override the `sortKey()` method.

9.22 2.12.7 (2010-06-13)

9.22.1 Bugs Fixed

- LP #142535: Fix faulty docstring for `manage_changeProperties` which incorrectly suggested that passing a simple dictionary as `REQUEST` argument was supported.
- LP #583702: Fixed regression in mailhost queue processor code introduced in 2.12.6 in the LP #574286 changes.
- Handle `Unauthorized` exceptions in `xmlrpc` correctly.
- Five's `processInputs()` would stomp on `:list` or `:tuple` values that contained ints or other non-strings, would clear out `:records` entirely, and would not do anything for `:record` fields.
- LP #143261: The (very old-fashioned) `Zope2.debug` interactive request debugger still referred to the toplevel module `Zope`, which was renamed to `Zope2` a long time ago.
- LP #142874: Naming objects `URL` or `URL1` broke several ZMI views.
- LP #142878: Remove `URL`-based suppression of access rules and site root objects. Suppression using `os.environ` still works.
- LP #143144: Fix documentation for the `zope.conf` `mount-point` directive.
- LP #142410: Do not index documents in a `KeywordIndex` if the document is missing the indexed attribute, if determining the value raises `AttributeError`, or if the indexed attribute is empty.
- LP #142590: The `DTMLMethod` and `DTMLDocument` `manage_edit` methods could not deal with `TaintedString` instances. Removed the entirely redundant `DTMLDocument.manage_edit` method at the same time.
- LP #142451: If non-recursive ownership changes are made using `changeOwnership`, do not touch any children.
- LP #142750 and LP #142481: To prevent confusion when choosing an `Id` and to avoid issues when creating two `VirtualHostMonsters` in the same container the `VirtualHostMoster` now has a default `Id`. It can no longer be selected, and the intermediary `Add` view is gone.

9.22.2 Features Added

- Updated packages:
 - `distribute` = 0.6.13
 - `five.formlib` = 1.0.3
 - `lxml` = 2.2.6
 - `Sphinx` = 0.6.6
 - `tempstorage` = 2.11.3
 - `zope.app.apidoc` = 3.6.6
 - `zope.app.applicationcontrol` = 3.5.1
 - `zope.app.authentication` = 3.6.1
 - `zope.app.i18n` = 3.6.2
 - `zope.app.intid` = 3.7.1
 - `zope.app.wsgi` = 3.6.1

- zope.contenttype = 3.4.3
- zope.container = 3.8.3
- zope.deferredimport = 3.5.2
- zope.intid = 3.7.2
- zope.principalannotation = 3.6.1
- Update Five's testbrowser to support both mechanize 0.1.x and 0.2.x.
- LP #142226: Added an extra keyword argument to the HTTPResponse setCookie method to suppress enclosing the cookie value field in double quotes.

9.23 2.12.6 (2010-05-17)

9.23.1 Restructuring

- Removed Zope2's own mkzeoinstance script in favor of the version from the new zope.mkzeoinstance package.

9.23.2 Features Added

- Updated packages:
 - DateTime = 2.12.2
 - distribute = 0.6.12
 - ExtensionClass = 2.13.1
 - initgroups = 2.13.0
 - Missing = 2.13.0
 - MultiMapping = 2.13.0
 - Persistence = 2.13.1
 - pytz = 2010h
 - Record = 2.13.0
 - RestrictedPython = 3.5.2
 - ThreadLock = 2.13.0
 - ZODB3 = 3.9.5
 - zope.app.locales = 3.6.1
 - zope.authentication = 3.7.1
 - zope.browser = 1.3
 - zope.cachedescriptors = 3.5.1
 - zope.deferredimport = 3.5.1
 - zope.i18n = 3.7.3
 - zope.i18nmessageid = 3.5.2
 - zope.lifecycleevent = 3.6.1

- zope.pagetemplate = 3.5.1
- zope.proxy = 3.6.0
- zope.security = 3.7.3

9.23.3 Bugs Fixed

- LP #142563: Fix `AccessControl.User.NullUnrestrictedUserTests.__str__`.
- Fix several template errors in `SiteErrorLog` (chameleon compatibility).
- LP #267820: Fix bad except clause in the `sequence_sort` method of the `<dtml-in>` tag.
- LP #351006: Don't nest block tags inside HTML `<p>` tags in `zExceptions.ExceptionFormatter`.
- LP #411837: Handle resource files with `.htm` extension properly, as page template resources.
- LP #435728: Fix indentation of `OFSP/help/sequence.py` docstring.
- LP #574286: Ensure that mailhosts which share a queue directory do not double-deliver mails, by sharing the thread which processes emails for that directory.
- `BaseRequest`: Fixed handling of errors in `'traverseName'`.

9.24 2.12.5 (2010-04-24)

9.24.1 Features Added

- Updated packages:
 - Acquisition = 2.13.3
 - distribute = 0.6.10
 - roman = 1.4.0
 - zope.annotation = 3.5.0
 - zope.app.cache = 3.6.0
 - zope.app.dav = 3.5.1
 - zope.app.i18n = 3.6.1
 - zope.app.localpermission = 3.7.2
 - zope.app.principalannotation = 3.7.0
 - zope.app.publication = 3.8.1
 - zope.app.publisher = 3.8.4
 - zope.app.renderer = 3.5.1
 - zope.app.security = 3.7.3
 - zope.app.session = 3.6.1
 - zope.app.testing = 3.7.3
 - zope.app.wsgi = 3.6.0
 - zope.app.zapi = 3.4.1

- zope.broken = 3.6.0
 - zope.catalog = 3.8.1
 - zope.formlib = 3.7.0
 - zope.i18nmessageid = 3.5.1
 - zope.index = 3.6.0
 - zope.keyreference = 3.6.2
 - zope.lifecycleevent = 3.6.0
 - zope.location = 3.6.0
 - zope.minmax = 1.1.2
 - zope.securitypolicy = 3.6.1
 - zope.viewlet = 3.6.1
- Downgrade the `manage_*` is discouraged. You should use event subscribers instead warnings to debug level logging. This particular warning hasn't motivated anyone to actually change any code.
 - Use the standard libraries `doctest` module in favor of the deprecated version in `zope.testing`.
 - LP #143013: make the maximum number of retries on `ConflictError` a configuration option.

9.24.2 Bugs Fixed

- `HTTPResponse`: Fixed handling of exceptions with unicode values.
- `zExceptions`: Fixed some unicode issues in `Unauthorized`.
- LP #372632, comments #15ff.: Fixed regression in `Unauthorized` handling.
- LP #563229: Process "evil" JSON cookies which contain double quotes in violation of RFC 2965 / 2616.
- Document `Products.PluginIndexes.PathIndex.PathIndex.insertEntry` as an API for use by subclasses.
- Fixed regression introduced in the resource directory fix in 2.12.4.
- LP #143655: don't prevent sorting using a path index.
- LP #142478: normalize terms passed to `PLexicon.queryLexicon` using the lexicon's pipeline (e.g., case flattening, stop word removal, etc.)
- Wrap exception views in the context of the published object.

9.25 2.12.4 (2010-04-05)

9.25.1 Restructuring

- Removed unmaintained build scripts for full Windows installers and `configure / make type builds`.
- Updated copyright and license information to conform with repository policy.
- Factored out `ZopeUndo` package into a separate distribution.

9.25.2 Features Added

- Updated packages:
 - Acquisition = 2.13.2
 - ExtensionClass = 2.13.0
 - Persistence = 2.13.0
 - pytz = 2010b
- There is now an event `ZPublisher.interfaces.IPubBeforeStreaming` which will be fired just before the first chunk of data is written to the response stream when using the `write()` method on the response. This is the last possible point at which response headers may be set in this case.

9.25.3 Bugs Fixed

- LP #142430: Avoid using the contexts `title_or_id` method in the `standard_error_message`.
- LP #257675: `request.form` contained `'-C:'` when no `QUERY_STRING` was in the environment.
- Zope 3-style resource directories would throw an `Unauthorized` error when trying to use `restrictedTraverse()` to reach a resource in a sub-directory of the resource directory.
- Restore ability to traverse to `'macros'` on template-based browser views.
- Protect `ZCTextIndex`'s `clear` method against storing `Acquisition` wrappers.
- LP #195761: fixed ZMI XML export / import and restored it to the UI.
- `MailHost` should fall back to `HELO` when `EHLO` fails.

9.26 2.12.3 (2010-01-12)

9.26.1 Bugs Fixed

- LP #491224: proper escaping of rendered error message
- LP #246983: Enabled unicode conflict resolution on variables inside `"string:"` expressions in `TALES`.
- Fixed possible `TypeError` while sending multipart emails.
- Also look for `ZEXP` imports within the `clienthome` directory. This provides a place to put imports that won't be clobbered by `buildout` in a `buildout`-based Zope instance.
- Fixed a `SyntaxError` in `utilities/load_site.py` script.

9.26.2 Features Added

- Made `OFS.Image.File` and `OFS.Image.Image` send `IObjectModifiedEvent` when created through their factories and modified through the ZMI forms (`manage_edit()` and `manage_upload()`).
- Moved `zope.formlib / zope.app.form` integration into a separate package called `five.formlib`.

9.27 2.12.2 (2009-12-22)

9.27.1 Features Added

- Updated packages:
 - ZODB3 = 3.9.4
 - docutils = 0.6
 - pytz = 2009r
 - zope.dottedname = 3.4.6
 - zope.i18n = 3.7.2
 - zope.interface = 3.5.3
 - zope.minmax = 1.1.1
 - zope.security = 3.7.2
 - zope.session = 3.9.2
 - zope.tal = 3.5.2
- Enhanced the internals of the `DateRangeIndex` based on an idea from `experimental.daterangeindexoptimisations`, thanks to Matt Hamilton.
- Updated the default value for `management_page_charset` from `iso-8859-1` to the nowadays more standard `utf-8`.
- Added `IPubBeforeAbort` event to mirror `IPubBeforeCommit` in failure scenarios. This event is fired just before `IPubFailure`, but, crucially, while the transaction is still open.
- Include bytes limited cache size in the cache parameters ZMI screen.
- Officially supporting Python 2.6 only (with unofficial support for Python 2.5) but dropping any support and responsibility for Python 2.4.

9.27.2 Bugs Fixed

- LP #143444: add labels to checkboxes / radio buttons on import / export form.
- LP #496941: Remove all mention of `standard_html_header` and `standard_html_footer` from default DTML content.
- Fixed a regression in `Products.PageTemplates` that meant filesystem templates using `Products.Five.browser.pagetemplatefile` would treat TALEX path expressions (but not python expressions) as protected code and so attempt to apply security. See original issue here: <http://codespeak.net/pipermail/z3-five/2007q2/002185.html>
- LP #491249: fix tabindex on ZRDB connection test form.
- LP #490514: preserve tainting when calling into DTML from ZPT.
- Avoid possible errors on test tear-down in `Products.Five.fiveconfigure`'s `cleanUp()` function if `Products.meta_types` has not been set

9.28 2.12.1 (2009-11-02)

9.28.1 Features Added

- Updated packages:
 - ZODB3 = 3.9.3 (fixes bug where blob conflict errors hung commits)
 - Acquisition = 2.12.4 (fixes problems with iteration support)
 - setuptools = 0.6c11
- LP #411732: Silence security declaration warnings for context and request on views protected by an interface.
- Assorted documentation cleanups, including a script to rebuild HTML documentation on Windows.
- Refactored Windows Service support to not need or use `zopeservice.py` in instances. This makes buildout-based instances work on Windows.

9.28.2 Bugs Fixed

- LP #440490: `zopectl fgldduser!run!debug` now work on Windows.
- LP #443005: `zopectl stop` works once more on Windows.
- LP #453723: `zopectl start` works again on non-Windows platforms.

9.29 2.12.0 (2009-10-01)

9.29.1 Features Added

- Updated packages:
 - ZODB3 = 3.9.0
- Backported clone of `ZopeVocabularyRegistry` from `zope.app.schema`, and sane registration of it during initialization of Five product.

9.29.2 Bugs Fixed

- Backported removal of experimental support for configuring the Twisted HTTP server as an alternative to `ZServer`.
- Backported fix for timezone issues in date index tests from trunk.
- LP #414757 (backported from Zope trunk): don't emit a `IEndRequestEvent` when clearing a cloned request.

9.30 2.12.0c1 (2009-09-04)

9.30.1 Features Added

- Updated packages:
 - Acquisition = 2.12.3

- pytz = 20091
- tempstorage = 2.11.2
- transaction = 1.0.0
- ZODB3 = 3.9.0c3
- zope.app.basic skin = 3.4.1
- zope.app.form = 3.8.1
- zope.component = 3.7.1
- zope.copypastemove = 3.5.2
- zope.i18n = 3.7.1
- zope.security = 3.7.1

9.30.2 Bugs Fixed

- Made the version information show up again, based on pkg_resources distribution information instead of the no longer existing version.txt.

9.31 2.12.0b4 (2008-08-06)

9.31.1 Features Added

- The send method of MailHost now supports unicode messages and email.Message.Message objects. It also now accepts charset and msg_type parameters to help with character, header and body encoding.
- Updated packages:
 - ZODB3 = 3.9.0b5
 - zope.testing = 3.7.7
- scripts: Added 'runzope' and 'zopectl' as entry points for instance scripts.

9.31.2 Bugs Fixed

- LP #418454: FTP server did not work with Python 2.6.X
- PythonScript: Fixed small Python 2.6 compatibility issue.
- mkzopeinstance: Made instance scripts more suitable for egg based installs. If you are using a customized skel, it has to be updated.
- Five: Fixed the permissions creation feature added in Zope 2.12.0a2.
- LP #399633: fixed interpreter paths
- MailHost manage form no longer interprets the value None as a string in user and password fields.

9.32 2.12.0b3 (2009-07-15)

9.32.1 Features Added

- Updated packages:
 - ZConfig = 2.7.1
 - ZODB = 3.9.0b2
 - pytz = 2009j
 - zope.app.component = 3.8.3
 - zope.app.pagetemplate = 3.7.1
 - zope.app.publisher = 3.8.3
 - zope.app.zcmlfiles = 3.5.5
 - zope.contenttype = 3.4.2
 - zope.dublincore = 3.4.3
 - zope.index = 3.5.2
 - zope.interface = 3.5.2
 - zope.testing = 3.7.6
 - zope.traversing = 3.7.1
- Added support to indexing datetime values to the PluginIndexes DateRangeIndex. The DateIndex already had this feature.

9.32.2 Restructuring

- PluginIndexes: Removed deprecated TextIndex.
- HelpSys now uses ZCTextIndex instead of the deprecated TextIndex. Please update your Zope databases by deleting the Product registrations in the Control Panel and restarting Zope.

9.32.3 Bugs Fixed

- LP #397861: exporting \$PYTHON in generated 'zopectl' for fixing import issue with "bin/zopectl adduser"
- PluginIndexes: Added 'indexSize' to IPluggableIndex.
- HelpSys: ProductHelp no longer depends on PluginIndexes initialization.
- App.Product: ProductHelp was broken since Zope 2.12.0a1.
- ObjectManagerNameChooser now also works with BTreeFolder2.
- Correctly handle exceptions in the ZPublisherExceptionHook.

9.33 2.12.0b2 (2009-05-27)

9.33.1 Restructuring

- Removed all use of `zope.app.pagetemplate` by cloning / simplifying client code.
- Use `zope.pagetemplate.engine` instead of `zope.app.pagetemplate.engine`. (update to versions 3.5.0 and 3.7.0, respectively, along with version 3.8.1 of `zope.app.publisher`).
- Use `IBrowserView` interface from `zope.browser.interfaces`, rather than `zope.publisher.interfaces.browser`.
- Use `IAdding` interface from `zope.browser.interfaces`, rather than `zope.app.container`.
- No longer depend on `zope.app.appsetup`; use the event implementations from `zope.processlifetime` instead.

9.33.2 Features Added

- `zExceptions.convertExceptionType`: new API, breaking out conversion of exception names to exception types from 'upgradeException'.
- Launchpad #374719: introducing new ZPublisher events: `PubStart`, `PubSuccess`, `PubFailure`, `PubAfterTraversal` and `PubBeforeCommit`.
- `Testing.ZopeTestCase`: Include a copy of `ZODB.tests.warnhook` to silence a `DeprecationWarning` under Python 2.6.
- Updated packages:
 - `python-gettext` 1.0
 - `pytz` 2009g
 - `zope.app.applicationcontrol` = 3.5.0
 - `zope.app.appsetup` 3.11
 - `zope.app.component` 3.8.2
 - `zope.app.container` 3.8.0
 - `zope.app.form` 3.8.0
 - `zope.app.http` 3.6.0
 - `zope.app.interface` 3.5.0
 - `zope.app.pagetemplate` 3.6.0
 - `zope.app.publication` 3.7.0
 - `zope.app.publisher` 3.8.0
 - `zope.browser` 1.2
 - `zope.component` 3.7.0
 - `zope.componentvocabulary` 1.0
 - `zope.container` 3.8.2
 - `zope.formlib` 3.6.0
 - `zope.lifecycleevent` 3.5.2

- zope.location 3.5.4
- zope.processlifetime 1.0
- zope.publisher 3.8.0
- zope.security 3.7.0
- zope.testing 3.7.4
- zope.traversing 3.7.0

9.33.3 Bugs Fixed

- Launchpad #374729: Encoding cookie values to avoid issues with firewalls and security proxies.
- Launchpad #373583: ZODBMountPoint - fixed broken mount support and extended the test suite.
- Launchpad #373621: catching and logging exceptions that could cause leaking of worker threads.
- Launchpad #373577: setting up standard logging earlier within the startup phase for improving the analysis of startup errors.
- Launchpad #373601: abort transaction before connection close in order to prevent connection leaks in case of persistent changes after the main transaction is closed.
- Fix BBB regression which prevented setting browser ID cookies from browser ID managers created before the HTTPOnly feature landed. <https://bugs.launchpad.net/bugs/374816>
- RESPONSE.handle_errors was wrongly set (to debug, should have been not debug). Also, the check for exception constructor arguments didn't account for exceptions that didn't override the `__init__` (which are most of them). The combination of those two problems caused the `standard_error_message` not to be called. Fixes <https://bugs.launchpad.net/zope2/+bug/372632>.
- DocumentTemplate.DT_Raise: use new 'zExceptions.convertExceptionType' API to allow raising non-builtin exceptions. Fixes <https://bugs.launchpad.net/zope2/+bug/372629>, which prevented viewing the "Try" tab of a script with no parameters.

9.34 2.12.0b1 (2009-05-06)

9.34.1 Restructuring

- No longer depend on `zope.app.locales`. Zope2 uses almost none of the translations provided in the package and is not required for most projects. The decision to include locales is left to the application developer now.
- Removed the dependency on `zope.app.testing` in favor of providing a more minimal placeless setup as part of ZopeTestCase for our own tests.
- updated to ZODB 3.9.0b1

9.34.2 Features Added

- `zExceptions.convertExceptionType`: new API, breaking out conversion of exception names to exception types from `upgradeException`.
- Extended `BrowserIdManager` to expose the `HTTPOnly` attribute for its cookie. Also via <https://bugs.launchpad.net/zope2/+bug/367393>.

- Added support for an optional `HTTPOnly` attribute of cookies (see <http://www.owasp.org/index.php/HTTPOnly>). Patch from Stephan Hofmockel, via <https://bugs.launchpad.net/zope2/+bug/367393>.

9.34.3 Bugs Fixed

- `ZPublisher.response.setBody`: don't append `Accept-Encoding` to `Vary` header if it is already present - this can make cache configuration difficult.

9.35 2.12.0a4 (2009-04-24)

9.35.1 Bugs Fixed

- fixed `versions.cfg` in order to support `zope.z2release` for creating a proper index structure

9.36 2.12.0a3 (2009-04-19)

The generated tarball for the 2.12.0a2 source release was incomplete, due to a `setuptools` and `Subversion` 1.6 incompatibility.

9.36.1 Restructuring

- Added automatic inline migration for databases created with older Zope versions. The `Versions` screen from the `Control_Panel` is now automatically removed on Zope startup.
- Removed more unused code of the versions support feature including the `Globals.VersionNameName` constant.

9.37 2.12.0a2 (2009-04-19)

9.37.1 Restructuring

- If the `<permission />` ZCML directive is used to declare a permission that does not exist, the permission will now be created automatically, defaulting to being granted to the `Manager` role only. This means it is possible to create new permissions using ZCML only. The permission will `Permissions` that already exist will not be changed.
- Using `<require set_schema="..." />` or `<require set_attributes="..." />` in the `<class />` directive now emits a warning rather than an error. The concept of protecting attribute 'set' does not exist in Zope 2, but it should be possible to re-use packages that do declare such protection.
- Updated to `Acquisition` 2.12.1.
- Updated to `DateTime` 2.12.0.
- Updated to `ZODB` 3.9.0a12.
- Removed the `getPackages` wrapper from `setup.py` which would force all versions to an exact requirement. This made it impossible to require newer versions of the dependencies. This kind of KGS information needs to be expressed in a different way.

- removed `extras_require` section from `setup.py` (this might possibly break legacy code).

9.37.2 Bugs Fixed

- Launchpad #348223: optimize catalog query by breaking out early from loop over indexes if the result set is already empty.
- Launchpad #344098: in `skel/etc/zope.conf.in`, replaced commented-out `read-only-database` option, which is deprecated, with pointers to the appropos sections of ZODB's `component.xml`. Updated the description of the `zserver-read-only-mode` directive to indicate its correct semantics (suppressing log / pid / lock files). Added deprecation to the `read-only-database` option, which has had no effect since Zope 2.6.
- “Permission tab”: correct wrong form parameter for the user-permission report
- PageTemplates: Made `PreferredCharsetResolver` work with new kinds of contexts that are not acquisition wrapped.
- Object managers should evaluate to `True` in a boolean test.

9.38 2.12.0a1 (2009-02-26)

9.38.1 Restructuring

- Switched `Products.PageTemplates` to directly use `zope.i18n.translate` and removed the `GlobalTranslationService` hook.
- Removed bridging code from `Product.Five` for `PlacelessTranslationService` and `Localizer`. Neither of the two is actually using this anymore.
- Removed the specification of `SOFTWARE_HOME` and `ZOPE_HOME` from the standard instance scripts. [hannosch]
- Made the specification of `SOFTWARE_HOME` and `ZOPE_HOME` optional. In addition `INSTANCE_HOME` is no longer required to run the tests of a source checkout of Zope.
- Removed the `test` command from `zopectl`. The `test.py` script it was relying on does no longer exist.
- Updated to ZODB 3.9.0a11. ZODB-level version support has been removed and `ZopeUndo` now is part of Zope2.
- The Zope2 SVN trunk is now a buildout pulling in all dependencies as actual released packages and not SVN externals anymore.
- Make use of the new `zope.container` and `zope.site` packages.
- Updated to newer versions of zope packages. Removed long deprecated `layer` and `skin` ZCML directives.
- Disabled the XML export on the UI level - the export functionality however is still available on the Python level.
- No longer show the `Help!` links in the ZMI, if there is no help available. The help system depends on the product registry.
- Updated the quick start page and simplified the standard content. The default `index.html` is now a page template.
- Removed deprecated `Draft` and `Version` support from `Products.OFSP`. Also removed version handling from the control panel. Versions are no longer supported on the ZODB level.
- Removed left-overs of the deprecated persistent product distribution mechanism.

- The persistent product registry is not required for starting Zope anymore. `enable-product-installation` can be set to off if you don't rely on the functionality provided by the registry.
- ZClasses have been deprecated for two major releases. They have been removed in this version of Zope.
- Avoid deprecation warnings for the md5 and sha modules in Python 2.6 by adding conditional imports for the hashlib module.
- Replaced imports from the 'Globals' module throughout the tree with imports from the actual modules; the 'Globals' module was always intended to be an area for shared data, rather than a "facade" for imports. Added `zope.deferred.deprecation` entries to 'Globals' for all symbols / modules previously imported directly.
- Protect against non-existing `zope.conf` path and products directories. This makes it possible to run a Zope instance without a Products or lib/python directory.
- Moved exception `MountedStorageError` from `ZODB.POSExceptions` to `Products.TemporaryFolder.mount` (now its only client).
- Moved Zope2-specific module, `ZODB/Mount.py`, to `Products/TemporaryFolder/mount.py` (its only client is `Products/TemporaryFolder/TemporaryFolder.py`).
- Removed spurious import-time dependencies from `Products/ZODBMountPoint/MountedObject.py`.
- Removed `Examples.zexp` from the skeleton. The TTW shopping cart isn't any good example of Zope usage anymore.
- Removed deprecated `ZTUtil.Iterator` module
- Removed deprecated `StructuredText` module
- Removed deprecated `TAL` module
- Removed deprecated modules from `Products.PageTemplates`.
- Removed deprecated `ZCML` directives from Five including the whole `Five.site` subpackage.

9.38.2 Features added

- `OFS.ObjectManager` now fully implements the `zope.container.IContainer` interface. For the last Zope2 releases it already claimed to implement the interface, but didn't actually full-fill the interface contract. This means you can start using more commonly used Python idioms to access objects inside object managers. Complete dictionary-like access and container methods including iteration are now supported. For each class derived from `ObjectManager` you can use for any instance `om`: `om.keys()` instead of `om.objectIds()`, `om.values()` instead of `om.objectValues()`, but also `om.items()`, `ob.get('id')`, `ob['id']`, `'id' in om`, `iter(om)`, `len(om)`, `om['id'] = object()` instead of `om._setObject('id', object())` and `del ob['id']`. Should contained items of the object manager have ids equal to any of the new method names, the objects will override the method, as expected in Acquisition enabled types. Adding new objects into object managers by those new names will no longer work, though. The added methods call the already existing methods internally, so if a derived type overwrote those, the new interface will provide the same functionality.
- Acquisition has been made aware of `__parent__` pointers. This allows direct access to many Zope 3 classes without the need to mixin Acquisition base classes for the security to work.
- `MailHost`: now uses `zope.sendmail` for delivering the mail. With this change `MailHost` integrates with the Zope transaction system (avoids sending dupe emails in case of conflict errors). In addition `MailHost` now provides support for asynchronous mail delivery. The 'Use queue' configuration option will create a mail queue on the filesystem (under 'Queue directory') and start a queue thread that checks the queue every three seconds. This decouples the sending of mail from its delivery. In addition `MailHosts` now supports encrypted connections through TLS/SSL.

- SiteErrorLog now includes the entry id in the information copied to the event log. This allows you to correlate a user error report with the event log after a restart, or let's you find the REQUEST information in the SiteErrorLog when looking at a traceback in the event log.

9.38.3 Bugs Fixed

- Launchpad #332168: Connection.py: do not expose DB connection strings through exceptions
- Specified height/width of icons in ZMI listings so the table doesn't jump around while loading.
- After the proper introduction of parent-pointers, it's now wrong to acquisition-wrap content providers. We will now use the "classic" content provider expression from Zope 3.
- Ported c69896 to Five. This fix makes it possible to provide a template using Python, and not have it being set to None by the viewlet manager directive.
- Made Five.testbrowser compatible with mechanize 0.1.7b.
- Launchpad #280334: Fixed problem with 'timeout' argument/attribute missing in testbrowser tests.
- Launchpad #267834: proper separation of HTTP header fields using CRLF as requested by RFC 2616.
- Launchpad #257276: fix for possible denial-of-service attack in PythonScript when passing an arbitrary module to the encode() or decode() of strings.
- Launchpad #257269: 'raise SystemExit' with a PythonScript could shutdown a complete Zope instance
- Switch to branch of 'zope.testbrowser' external which suppresses over-the-wire tests.
- Launchpad #143902: Fixed App.ImageFile to use a stream iterator to output the file. Avoid loading the file content when guessing the mimetype and only load the first 1024 bytes of the file when it cannot be guessed from the filename.
- Changed PageTemplateFile not to load the file contents on Zope startup anymore but on first access instead. This brings them inline with the zope.pagetemplate version and speeds up Zope startup.
- Collector #2278: form ':record' objects did not implement enough of the mapping protocol.
- "version.txt" file was being written to the wrong place by the Makefile, causing Zope to report "unreleased version" even for released versions.
- Five.browser.metaconfigure.page didn't protect names from interface superclasses (<http://www.zope.org/Collectors/Zope/2333>)
- DAV: litmus "notowner_modify" tests warn during a MOVE request because we returned "412 Precondition Failed" instead of "423 Locked" when the resource attempting to be moved was itself locked. Fixed by changing Resource.Resource.MOVE to raise the correct error.
- DAV: litmus props tests 19: propvalnspace and 20: propwformed were failing because Zope did not strip off the xmlns: attribute attached to XML property values. We now strip off all attributes that look like xmlns declarations.
- DAV: When a client attempted to unlock a resource with a token that the resource hadn't been locked with, in the past we returned a 204 response. This was incorrect. The "correct" behavior is to do what mod_dav does, which is return a '400 Bad Request' error. This was caught by litmus locks.notowner_lock test #10. See <http://lists.w3.org/Archives/Public/w3c-dist-auth/2001JanMar/0099.html> for further rationale.
- When Zope properties were set via DAV in the "null" namespace (xmlns='') a subsequent PROPFIND for the property would cause the XML representation for that property to show a namespace of xmlns="None". Fixed within OFS.PropertySheets.dav__propstat.
- integrated theuni's additional test from 2.11 (see r73132)

- Relaxed requirements for context of `Products.Five.browser.pagetemplatefile.ZopeTwoPageTemplateFile`, to reduce barriers for testing renderability of views which use them. (<http://www.zope.org/Collectors/Zope/2327>)
- `PluginIndexes`: Fixed 'parseIndexRequest' for false values.
- Collector #2263: 'field2ulines' did not convert empty string correctly.
- Collector #2198: Zope 3.3 fix breaks Five 1.5 `test_getNextUtility`
- Prevent `ZPublisher` from inserting incorrect `<base/>` tags into the headers of plain html files served from Zope3 resource directories.
- Changed the condition checking for setting status of `HTTPResponse` from to account for new-style classes.
- The `Wrapper_compare` function from `tp_compare` to `tp_richcompare`. Also another function `Wrapper_richcompare` is added.
- The doc test has been slightly changed in `ZPublisher` to get the error message extracted correctly.
- The changes made in `Acquisition.c` in Implicit Acquisition comparison made avail to Explicit Acquisition comparison also.
- `zopedoctest` no longer breaks if the URL contains more than one question mark. It broke even when the second question mark was correctly quoted.

9.38.4 Other Changes

- Added `lib/python/webdav/litmus-results.txt` explaining current test results from the litmus WebDAV torture test.
- `DocumentTemplate.DT_Var.newline_to_br()`: Simpler, faster implementation.