

# Indico 1.2 Installation for Ubuntu

---

Daniel Gordon  
Naval Research Laboratory  
Plasma Physics Division  
August 11, 2015

**Disclaimer:** This is not an official Naval Research Laboratory document. Neither Daniel Gordon nor the U.S. Government make any guarantees or assume any liability whatsoever in connection with the use of this document.

## Typesetting Convention

- Literal names appearing in the file system, or command line entries, are given in bold courier, e.g., **/usr** and **ls**.
- Names that are meant to be substituted with something appropriate to a particular installation are shown in red, e.g., **myhost.mydomain**.

## Overview

The indico software is a python program that talks to an Apache web server using WSGI. This requires a web server module, known as mod\_wsgi. Indico also relies on other technologies that have to be installed separately. At the time of this writing, the complexity of installation increases with the indico version number.

The indico software comes in two major varieties, versions up to 1.2 use only ZOEPE database technology. The developers decided at that point to start gradually switching to SQLAlchemy database technology, with the goal of releasing a version 2.0 that uses SQLAlchemy only (hopefully). At the time of this writing the latest version is 1.9.4, which still appears to require both ZOEPE and SQLAlchemy.

Both the ZOEPE and SQLAlchemy systems have two basic software components, a database server daemon and a library for talking to the server from within python. In the case of ZOEPE, the server technology is called ZEO (ZOEPE Enterprise Objects), and the actual code daemon is called **runzeo**. In the case of SQLAlchemy, the recommended technology is postgresSQL. The code daemon is called **postgres** on Ubuntu.

The indico software also relies on daemons for managing scheduled tasks. In version 1.2 and prior, the technology is called redis, and the code daemon is called **redis-server**. In later versions, a further technology called Celery is introduced. The celery server uses the redis-server as a “broker.”

*In this guide we will use version 1.2 which simplifies installation considerably.*

One major lesson is that indico should be installed in a **virtualenv**. The **virtualenv** is a technology for creating an isolated python environment with a minimal set of modules that are needed for a given application. This is important because the indico software has an extremely delicate dependency chain. A working indico installation can be easily broken by installing a new python module. Moreover, installing indico into an existing python installation can break other python components, including critical package management programs like **pip**.

*Once the indico server is up do not install anything else into the indico virtualenv.*

Additional python modules that are not directly called from the indico API (e.g., **maildump**) should be installed into the system environment, or a separate **virtualenv**.

## Useful Linux Information

### Installing Packages

Package managers allow software to be installed under an umbrella that automatically accounts for dependencies on other software. The system that is used on Ubuntu is called Debian. The command to install a package is **apt-get install somepackage**. The package management system maintains a database, which has to be manually updated at certain times using **apt-get update**. If an installation fails this may resolve the issue. Searching for a package is supported using **apt-cache search somepackage**. If a package cannot be found, it may be necessary to edit the apt configuration files, so that they point to additional repositories on the internet.

Python packages can sometimes be installed via **apt-get**, but sometimes must be installed using one of three package management system just for Python. The most basic is distutils, which requires setting up a source directory and executing **python setup.py install**. The second is Easy Install, which retrieves a package from the internet using the command **easy\_install somepackage**. Later, **easy\_install** was found to have some flaws (esp. lack of uninstall), so **pip** was introduced, which is invoked as **pip install somepackage**.

### Root Access

On Ubuntu, root login is disabled. Therefore most installation commands have to be entered via **sudo**. Prepending any command with **sudo** runs the command as the root user, after entering the root password. However, this is not always enough, because we may need to carry environment variables, including the path, into whatever command is being issued. Although it is not necessary in every case, to simplify the process, it is recommended that all commands be prepended with:

```
sudo -EH env "PATH=$PATH"
```

In a few instances, this is critical, and if not done properly libraries might be installed in the wrong place.

### File Permissions

Successful operation of Indico is sensitive to file permissions. Rather than changing the permissions, it is often better to change the ownership from root to the Apache user, which on Ubuntu should be **www-data**. Permissions errors will be apparent from the log files **/var/log/apache2/error.log** and **/opt/indico/log/indico.log**. Determine the full path of the offending directory and execute

```
sudo chown -R www-data:www-data the_directory_full_path
```

The option **-R** causes the ownership of everything in the directory to also be changed.

## Install Ubuntu Server 14.04

- ❖ Burn the DVD or create the bootable USB stick and run it
  - You may need to enter the setup program to set the boot order, usually by holding the "del" key while the computer is starting up.
- ❖ Most inputs into the installer are straightforward
- ❖ Configure the network manually as directed by the network administrator
- ❖ Add the following options
  - OpenSSH server
  - Mail server
    - Select internet site
    - Use the defaults
- ❖ Upon reboot you will get a text interface only. Check network connectivity with **ping 0.0.0.0** (substitute a suitable IP address). If it fails, you may not have had network connectivity during installation. Fix that and start over.
- ❖ If a desktop environment is desired this must be installed separately. Alternatively, one can use Ubuntu desktop 14.04, possibly with some performance penalty

## Set up the python virtualenv

- ❖ **apt-get update**
- ❖ **apt-get install python-pip**
- ❖ **pip install virtualenv**
- ❖ **mkdir /usr/local/pythonenv**

- ❖ **cd /usr/local/pythonenv**
- ❖ **virtualenv BASELINE**
- ❖ **virtualenv indico12**
- ❖ **source indico12/bin/activate**
  - In this case do not use sudo
  - This exports a new PATH variable, and must be repeated if you start a new terminal

## Install Indico Libraries

- ❖ Install required and useful system libraries (cf. indico docs)
  - **apt-get install libxslt-dev**
  - **apt-get install libffi-dev**
  - **apt-get install libldap2-dev**
  - **apt-get install libsasl2-dev**
  - **apt-get install lib32z1-dev**
  - **apt-get install Cython**
  - **apt-get install git**
  - **apt-get install redis-server**
- ❖ To install the python libraries there are two ways:
  - Install python libraries with easy\_install:
    - **pip install python-ldap**
    - **easy\_install indico**
  - Build from source (only do this if easy install fails for some reason)
    - **cd ~**
    - **git clone <https://github.com/indico/indico>**
    - **cd indico**
    - **git checkout v1.2**
    - **pip install python-ldap**
    - **pip install -r requirements.dev.txt**
    - **pip install -r requirements.txt**
    - **fab setup\_deps**
    - **python setup.py install**

## Setup indico database and web site

- ❖ **indico\_initial\_setup**
- ❖ **cd /opt/indico/etc**
- ❖ Edit zdctl.conf
  - Give the full path for runzeo, e.g.,  
**/usr/local/pythonenv/indico12/bin/runzeo**
  - Make sure username is correct. On Ubuntu it should be **www-data**. If in doubt you can check **/etc/group** and make sure the assumed username appears there.

- ❖ Start the ZEO server daemon
  - **zdaemon -C zdctl.conf start**
  - If a **permission denied** error is encountered, involving **namespace\_packages.txt**, change the ownership, or simply delete that particular file.
  - Verify that the server is running
    - **ps -C runzeo**
    - You should see a process called **runzeo**
- ❖ The redis-server should have started upon being installed. Check to see if it is running:
  - **ps -C redis-server**
  - You should see a process called **redis-server**
- ❖ Copy **/opt/indico/etc/indico.conf** from appendix A. Enter the server name under the **URLs** heading, configure the settings under the **SMTP** heading, and configure the addresses under the **EMAIL ADDRESSES** heading.

## Configure Apache Web Server

- ❖ **apt-get install apache2**
  - The server should be automatically started. At any time the server may be started, stopped, or restarted using **apache2ctl start**, **apache2ctl stop**, and **apache2ctl restart**.
- ❖ **apt-get install libapache2-mod-wsgi**
  - The module should be automatically enabled. This can be checked by looking for **wsgi.conf** in **/etc/apache2/mods-enabled**. If not present run
    - **a2enmod wsgi**
- ❖ The ssl module should already be installed and enabled. This can be checked by looking for **ssl.conf** in **/etc/apache2/mods-enabled**. If not present run
  - **a2enmod ssl**
- ❖ Edit **/etc/apache2/apache2.conf**
  - Change the Directory block pointing at **/var/www/** to point at **/opt/indico/htdocs/**
  - Add the the following lines
    - **WSGIPythonHome /usr/local/pythonenv/BASELINE**
    - **WSGIPythonPath /usr/local/pythonenv/indico12/lib/python2.7/site-packages**
    - **WSGIPythonEggs /opt/indico/tmp/egg-cache**
- ❖ Create the directory **/opt/indico/tmp/egg-cache** if necessary and change ownership to the apache user, probably **www-data**
- ❖ Create and edit the file **/etc/apache2/sites-available/a2indico.conf**
  - Refer to appendix B for contents of file
  - Enable the site using **a2ensite a2indico**
- ❖ Create self-signed certificate and key referenced in **a2indico.conf**

- ❖ Modify **default-ssl.conf** to point to the same certificate and key as **a2indico.conf**
- ❖ **a2ensite default-ssl**
- ❖ **apache2ctl restart**

## Setup Background Tasks and Finish Up

- ❖ In the following we assume the apache user and group are **www-data**
- ❖ Start the task scheduler
  - **sudo -u www-data -EH env "PATH=\$PATH" indico\_scheduler start**
  - You may have to change ownership of the contents of an **EGG-INFO** directory to **www-data**, see the path given in the **permission denied** error message if it appears.
  - You may have to create **/var/www/.python-eggs** and set the ownership to **www-data**. Or, set a different **PYTHON\_EGG\_CACHE** upon **www-data** login.
- ❖ **Verify that all the servers are running**
  - **ps -C indico\_schedule**
  - **ps -C apache2**
  - **ps -C redis-server**
  - **ps -C runzeo**

## Create Admin Account

From your web browser, access <https://myserver.mydomain/indico>, and press the orange Login button. If this produces an error, try the following:

- ❖ Close the browser
- ❖ **python -c "import bcrypt"**
- ❖ **apache2ctl restart**

Try to login again. After entering your information, you are supposed to get an e-mail for authentication purposes. Note that depending on your network configuration, the e-mail address might have to be an internal one. To create an administrator from the command line use **indico\_admin create**.

## Starting Indico Following System Reboot

Upon rebooting the server, or after a power failure, both apache and the redis-server should restart automatically. To restart the ZEO daemon and the scheduler daemon execute the following:

```
❖ source /usr/local/pythonenv/indico12/bin/activate
❖ sudo -EH env "PATH=$PATH" zdaemon -C
  /opt/indico/etc/zdctl.conf start
❖ sudo -EH -u www-data env "PATH=$PATH" indico_scheduler -f
  start
```

## Appendix A : Indico Configuration File

Contents of /opt/indico/etc/indico.conf follows

```
#-----
# Indico configuration file
#-----
#
# This file is read on Apache startup. If you change any value you will need to
# restart Apache afterwards to have its changes take effect.
#
# Lines starting with "#" are comments and they will be ignored by Indico.

#-----
# ZODB
#-----

DBConnectionParams = ('localhost', 9675)
DBUserName          = ""
DBPassword          = ""
DBRealm             = ""

#-----
# REDIS
#-----

# To enable redis, specify a valid redis connection string here.
# Example: redis://unused:password@localhost:6379/0
# You also need to install the python redis client (pip install redis hiredis)
# Note that the Redis server needs to run at least Redis 2.6 with LUA support.
RedisConnectionURL = 'redis://localhost:6379/0'

#-----
# SECURITY
#-----
# Possible values of SanitizationLevel:
# 0: Escape all HTML tags
# 1: Raise error if styles are used
# 2: Raise error if SCRIPT tag is used
# 3: No input filtering is done (DANGEROUS!)

SanitizationLevel = 2

# AuthenticatedEnforceSecure controls whether HTTPS should be enforced for
# authentication and while logged in.
```

```

AuthenticatedEnforceSecure = "yes"

# Possible values of CSRFLevel:
# 0: Disable all CSRF checks (DANGEROUS!)
# 1: Check CSRF token for service requests
# 2: Check CSRF token for service requests and cookie-based API requests
# 3: The above & require an empty or valid referer for normal POST requests
CSRFLevel = 2

#-----
# Development
#-----
# You can force a certain number of (simulated) database conflicts on every
# by setting ForceConflicts to the number of conflict errors you want:
#ForceConflicts = 0
#
# You can cause all error-like exceptions (i.e. not something like AccessError)
# to be propagated outside the Indico WSGI app by enabling this option. This has
# the advantage of triggering the Werkzeug debugger of the embedded server even
# in case of e.g. a MaKaCError.
#PropagateAllExceptions = False

#-----
# URLs
#-----
# BaseURL is the url where Indico will be accessed:

BaseURL = "http://myserver.mydomain:80/indico"

# BaseSecureURL is the HTTPS location of Indico. Set empty string if you want to use
# http.

BaseSecureURL = "https:// myserver.mydomain:443/indico"

# If you are using a custom user authentication system (see below) uncomment
# the following lines and specify the new URLs.
#
# LoginURL = ""
# RegistrationURL = ""

# Support the old mod_python-style URLs (conferenceDisplay.py?confId=123 etc.) by
# redirecting
# them to their new URL.
#RouteOldUrls = False

#-----
# DIRECTORIES
#-----
ArchiveDir = "/opt/indico/archive"
BinDir = "/opt/indico/bin"
ConfigurationDir = "/opt/indico/etc"
DocumentationDir = "/opt/indico/doc"
HtdocsDir = "/opt/indico/htdocs"
LogDir = "/opt/indico/log"
UploadedFilesTempDir = "/opt/indico/tmp"
XMLCacheDir = "/opt/indico/cache"
# You can use 'redis', 'memcached' or 'files'. The files backend caches objects in
# XMLCacheDir while the Memcached backend uses one or more memcached servers.

```



```

#CacheBackend          = 'files'
# When using memcached, provide a tuple containing 'ip:port' entries for the
# memcached servers. Do not forget to firewall those servers as memcached has no
# authentication at all!
#MemcachedServers      = ('127.0.0.1:11211',)
# When using redis, provide a redis connection string for the Redis server.
#RedisCacheURL = 'redis://unused:password@localhost:6379/1'

#-----
# SMTP
#-----
SmtpServer              = (mysmtp.mydomain, 25)
SmtpLogin               = ""
SmtpPassword           = ""

# Unless you are using an AUTH server leave login and password empty
# If your SMTP server is using TLS write "yes", otherwise write "no"

SmtpUseTLS              = "no"

#-----
# EMAIL ADDRESSES
#-----
# SupportEmail is the email address where all automatically generated
# application errors will be sent to.

SupportEmail            = "myaddress@mydomain"

# PublicSupportEmail is an email address that will be shown in Indico and where
# users are expected to find help when they have using the website.

PublicSupportEmail     = "myaddress@mydomain"

# NoReplyEmail is the email address showed when we do not want the users to answer
# an automatically generated email.

NoReplyEmail           = "noreply.root@mydomain"

#-----
# FILE UPLOAD
#-----
# Here you can limit the maximum size of all the uploaded files (in MB) in a
# request
# default: 0 (unlimited)

MaxUploadFilesTotalSize = 0

# Here you can limit the maximum size of an uploaded file (in MB)
# default: 0 (unlimited)

MaxUploadFileSize = 0

#-----
# FILE CONVERSION
#-----
# Indico has an interface to interact with an external file conversion system
# to convert from some formats to others but right now there is no publicly
# available file conversion software.

```

```

#
# If you are interested in this feature please contact us at:
#   indico-team@cern.ch
#
# FileConverter      = {"conversion_server": "localhost", "response_url":
"http://localhost/conversion-finished"}

#-----
# STATIC FILE DELIVERY
#-----
# Indico supports the use of the X-Sendfile and X-Accel-Redirect headers:
#
# http://blog.lighttpd.net/articles/2006/07/02/x-sendfile
# http://wiki.nginx.org/X-accel
#
# If your webserver supports this feature and you want to activate it,
# you should enable it here
#
# X-Sendfile (apache with mod_xsendfile, lighttpd):
# StaticFileMethod = 'xsendfile'
#
# X-Accel-Redirect (nginx):
# StaticFileMethod = ('xaccelredirect', {
#     '/opt/indico/archive': '/.xsf/archive',
#     '/opt/indico/cache': '/.xsf/cache',
#     '/opt/indico/htdocs': '/.xsf/htdocs'
# })
# Because of the way nginx works (using URLs instead of paths) you also need to map the
.xsf urls to
# the paths in your nginx config (for each entry in the dict above):
# location /.xsf/archive/ {
#     internal;
#     alias /opt/indico/archive/;
# }
# DO NOT forget the "internal;" statement - it prevents users from accessing those URLs
directly.

#StaticFileMethod = None

#-----
# AUTHENTICATION
#-----
# Indico ships with a fully working user authentication system (Local) but you
# can use an external system.
#
# If you want to have all users authenticate only to an external user management
# system (for example: MyAuthSystem) you should write:
#
#     AuthenticatorList = [('MyAuthSystem', {})]
#
# Where the second element of the tuple is a dictionary with the configuration.
#
# If you want to authenticate users against more than one authentication system
# you can write them like this (systems will be queried about users from left
# to right):
#
#     AuthenticatorList = [('Local', {}), ('LDAP', {}), ('MyAuthSystem', {})]
#

```

```

# If you want to setup SSO in your authenticator you should setup the variable
# SSOActive to True.
#
# AuthenticatorList = [('MyAuthSystem', {'SSOActive': True})]
#
# You can also can configure the mapping of the SSO attributes with the variable
SSOMapping. An example:
#
# AuthenticatorList = [ ('Local', {}),
#                       ('MyAuthSystem', {'SSOActive': True,
#                                          'LogoutCallbackURL':
'https://example.com/wsignout',
#                                          'SSOMapping': { 'email': 'ADFS_EMAIL',
#                                                        'login': 'ADFS_LOGIN',
#                                                        'personId': 'ADFS_PERSONID',
#                                                        'phone': 'ADFS_PHONENUMBER',
#                                                        'fax': 'ADFS_FAXNUMBER',
#                                                        'lastname': 'ADFS_LASTNAME',
#                                                        'firstname':
'ADFS_FIRSTNAME',
#                                                        'institute':
'ADFS_HOMEINSTITUTE'},
#                                          'ResetPasswordMessage': 'Please contact
helpdesk or access this url: http://example.com'
#                                          })]
#
#
# Please note that some authentication systems may require extra dependency libs.
# For instance, for LDAP authentication, you will also need to install the python-ldap
lib:
#
# $ pip install python-ldap
#
#
# As an example if you want to setup the LDAP Authenticator you should define as
following:
#
# AuthenticatorList = [('LDAP', {# by default the port is 389 if you do not specify.
#                               'uri': '[ldap|ldaps]://myldapservers.example.com:port',
#                               # use TLS (do so, if your server allows)
#                               'useTLS': True,
#                               # filter parameters for users, base DN to use
#                               'peopleDNQuery': ('uid={0}', 'DC=example,DC=com'),
#                               # filter parameters for groups, base DN to use
#                               'groupDNQuery': ('cn={0}',
'OU=Groups,DC=example,DC=com'),
#                               # SLAPD or ActiveDirectory
#                               'groupStyle': 'SLAPD',
#                               # access credentials of a user with read access
#                               'accessCredentials':
('CN=user,OU=Users,DC=example,DC=com', 'secret_password')
#                               # custom user filter
#                               'customUserFilter': "(objectCategory=user)"
#                               })]
#

```

```

#
#
# The default configuration will use only Indico's authentication system.

AuthenticatorList    = [('Local', {})]

# If you only want to allow users to login via SSO you can disable the normal login page.
# DisplayLoginPage = True

# Sessions are only stored for a certain time. You can modify the duration here. By
setting
# the lifetime to 0 the cookie will expire when the browser is closed.
# SessionLifetime = 86400 * 31

#-----
# OFFLINE WEBSITE PACKAGES CREATION
#-----
# Indico allows users to download their event so they can run it offline.
#
# The OfflineStore variable points to a directory that Indico will use to store
# the offline website packages.
#
# If the OfflineStore variable is not set, the functionality will be disabled.
#
# OfflineStore          = "/opt/indico/archive"

#-----
# LATEX/PDF GENERATION
#-----
# Set the path to pdflatex if it's not within PATH:
# PDFLatexProgram = 'pdflatex'
#
# In strict mode, a non-zero status code is considered failure. Disable this if
# you have old contributions that contain invalid LaTeX and you prefer possibly
# weird-looking PDFs over an error and no PDF at all.
# StrictLatex = True

#-----
# OAUTH PARAMETERS
#-----

# Time to live of the OAuth Tokens
OAuthAccessTokenTTL = 10000

#-----
# INDICO MOBILE
#-----

# If you have an installation of the mobile version of Indico, you can enable the
# notification that encourages its usage when the user accesses the Desktop Indico
# from a mobile device. One only needs to add the URL of Indico mobile here:
# MobileURL = "http://m.indico.your.domain"

```

## Appendix B : Indico Virtual Host Configuration

Contents of `/etc/apache2/sites-available/a2indico.conf` follows

```
AddDefaultCharset UTF-8
```

```
<VirtualHost *:80>
```

```
    ErrorLog /var/log/apache2/error.log
    LogLevel warn
```

```
    Alias /indico/images "/opt/indico/htdocs/images"
    Alias /indico/css "/opt/indico/htdocs/css"
    Alias /indico/js "/opt/indico/htdocs/js"
    Alias /indico/ihelp "/opt/indico/htdocs/ihelp"
```

```
    WSGIDaemonProcess WSGIDAEMON processes=32 threads=1 inactivity-timeout=3600 maximum-
requests=10000 python-path=/usr/local/pythonenv/indico12/lib/python2.7/site-packages
python-eggs=/opt/indico/tmp/egg-cache
```

```
    WSGIScriptAlias /indico "/opt/indico/htdocs/indico.wsgi"
```

```
    <Directory "/opt/indico">
        WSGIProcessGroup WSGIDAEMON
        WSGIApplicationGroup %{GLOBAL}
        AllowOverride None
        Options FollowSymLinks
        Order deny,allow
        Allow from all
```

```
    </Directory>
```

```
</VirtualHost>
```

```
<VirtualHost *:443>
```

```
    ErrorLog /var/log/apache2/error.log
    LogLevel warn
```

```
    Alias /indico/images "/opt/indico/htdocs/images"
    Alias /indico/css "/opt/indico/htdocs/css"
    Alias /indico/js "/opt/indico/htdocs/js"
    Alias /indico/ihelp "/opt/indico/htdocs/ihelp"
```

```
    WSGIScriptAlias /indico "/opt/indico/htdocs/indico.wsgi"
```

```
    <Directory "/opt/indico">
        WSGIProcessGroup WSGIDAEMON
        WSGIApplicationGroup %{GLOBAL}
        AllowOverride None
        Options FollowSymLinks
        Order deny,allow
        Allow from all
```

```
    </Directory>
```

```
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/mycert.crt
    SSLCertificateKeyFile /etc/ssl/certs/mykey.key
```

```
</VirtualHost>
```