**Hướng dẫn cài đặt Bugzilla trên Windows**

Bạn có thể tham khảo tại đây <https://wiki.mozilla.org/Bugzilla:Win32Install>

Thông thường thì Bugzilla được cài đặt trên Linux và Solaris. Để cài đặt Bugzilla trên hệ điều hành khác thì bạn cần phải xem xét tới các chú ý cần thiết đối với việc cài đặt trên hệ điều hành đó. Sau đây là hướng dẫn về việc cài đặt Bugzilla trên hệ điều hành Windows.

Việc cài đặt và vận hành Bugzilla trên Windows khó khăn hơn rất nhiều so với trên Unix. Vì vậy, tốt nhất chúng tôi khuyên rằng bạn nên cài đặt và làm việc với Bugzilla trên các hệ thống Unix (GNU/Linux).

Sau đây chúng ta sẽ tiến hành cài đặt (đối với phiên bản Bugzilla 4.4):

1. Cài đặt Perl (5.8.1 hoặc cao hơn (ở đây chúng ta sẽ sử dụng bản 5.14), bản 5.16 không hỗ trợ Template-Toolkit, xem thêm tại <http://code.activestate.com/ppm/Template-Toolkit/>.)
2. Cài đặt Database Engine
3. Cài đặt Webserver
4. Cài đặt Bugzilla
5. Cài đặt các module của Perl
6. Cài đặt Mail Transfer Agent (Sendmail 8.7 hoặc cao hơn, hoặc một MTA tương thích với phiên bản này)
7. Cấu hình

**Cài đặt Perl**

Nếu trên máy tính của bạn không có Perl và hệ điều hành của bạn không cung cấp sẵn gói này thì bạn có thể tải Perl về và cài đặt tại <http://www.perl.org>.

**Cài đặt Database Engine**

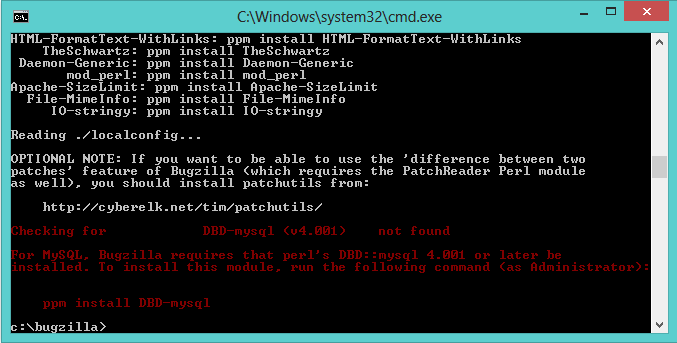
Bugzilla hỗ trợ các database server: MySQL, PostgreSQL và Oracle. Bạn chỉ cần một trong các hệ thống trên để sử dụng Bugzilla.

**Cài đặt MySQL**

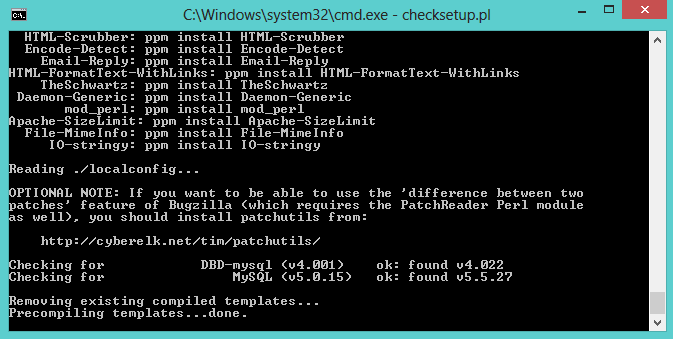
Bạn có thể truy cập <http://www.mysql.com> để tải MySQL. Bạn cần MySQL phiên bản 5.0.15 hoặc cao hơn.

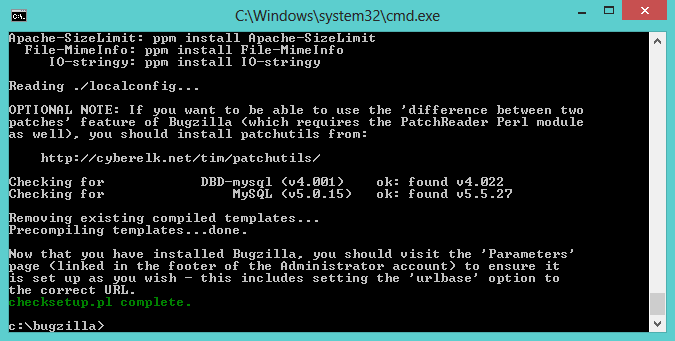
Nếu bạn cài đặt từ hệ thống packaging/installation khác như là .rpm (RPM Package Manager), .deb (Debian Package), .exe (Windows Executeable), hoặc .msi (Windows Installer) thì bạn phải chắc chắn rằng MySQL đã được start khi máy bạn khởi động.

Nếu sau khi bạn checksetup mà có thông báo như thế này thì bạn hãy tiến hành cài đặt DBD-mysql:



Bạn cài đặt DBD-mysql bằng lệnh hoặc bằng Perl Package Manager. Sau khi cài xong bạn chạy lại checksetup.pl ta được như sau:





**Cài đặt PostgreSQL**

Bạn cần PostgreSQL phiên bản 8.03.0000 hoặc cao hơn. Bạn có thể truy cập vào <http://www.postgresql.org> để tải PostgreSQL.

Nếu bạn cài đặt từ hệ thống packaging/installation khác như là .rpm (RPM Package Manager), .deb (Debian Package), .exe (Windows Executeable), hoặc .msi (Windows Installer) thì bạn phải chắc chắn rằng PostgreSQL đã được start khi máy bạn khởi động.

**Cài đặt Oracle**

Đầu tiên bạn phải log in vào DB của mình. Bạn cần Oracle phiên bản 10.02.0 hoặc cao hơn

Nếu bạn cài đặt từ hệ thống packaging/installation khác như là .rpm (RPM Package Manager), .deb (Debian Package), .exe (Windows Executeable), hoặc .msi (Windows Installer) thì bạn phải chắc chắn rằng Oracle đã được start khi máy bạn khởi động.

**Cài đặt Web Server**

Installed Version Test: view the default welcome page at http://<your-machine>/

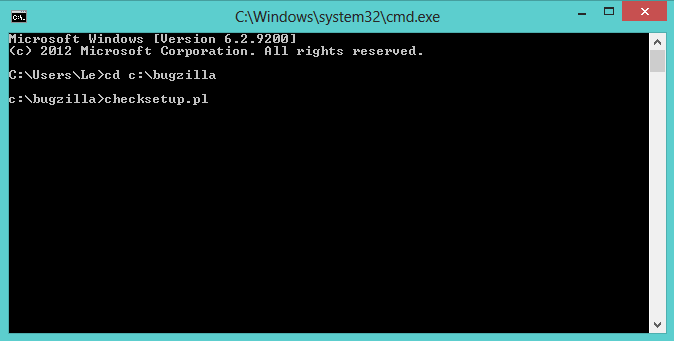
Hầu như các web server có khả năng chạy các CGI scipt. Tuy nhiên, chúng tôi khuyên sử dụng Apache web server (1.3.x hoặc 2.x). Bạn có thể truy cập vào <http://httpd.apache.org> để tải apache.

**Cài đặt Bugzilla**

Bạn có thể tải Bugzilla tại <http://www.bugzilla.org/download/> hoặc <https://wiki.mozilla.org/Bugzilla:Bzr> và đặt nó vào thư mục thích hợp mà web server có thể sử dụng được (“apache” hoặc “www”). Trong hướng dẫn này chúng ta sẽ cài đặt Bugzilla vào thư ổ đĩa C. Khi đó đường dẫn Bugzilla sẽ là C:\bugzilla.

**Cài đặt các Module của Perl**

Tiến trình cài đặt của Bugzilla bạn gọi script checksetup.pl. Đầu tiên nó sẽ kiểm tra xem bạn có phiên bản thích hợp của các module được yêu cầu hay không.



checksetup.pl sẽ in ra một danh sách các module không bắt buộc và các module được yêu cầu của Perl, cùng với version nếu cài nhiều phiên bản trên máy tính của bạn. Danh sách các module yêu cầu khá dài tuy nhiên bạn có thể đã cài đặt một vài module rồi.

Bạn có thể cài đặt các module còn thiếu của Perl bằng cách sử dụng package manager được cung cấp bởi hệ điều hành của bạn (ví dụ “rpm” hoặc “yum” trên Linux, hoặc “ppm” trên Windows nếu sử dụng ActivePerl).

Một số module còn thiếu hoặc quá cũ thì chúng tôi khuyên bạn sử dụng script install-module.pl (không làm việc với ActivePerl trên Windows).

**Chú ý:** nếu bạn đang sử dụng hệ thống package-based, và bạn cài đặt các module Perl từ CPAN, bạn cần phải cài đặt các gói “development” cho MySQL và GD trước khi cài đặt các module Perl liên quan. Tên của những gói này sẽ thay đổi tùy theo sự phân bố mà bạn sử dụng, như thường thì nó được gọi là <packagename>-devel.

Dưới đây là danh sách các module:

Required Perl modules:

1. CGI (3.51)
2. Date::Format (2.23)
3. DateTime (0.28)
4. DateTime::TimeZone (0.71)
5. DBI (1.54)
6. DBD::mysql (4.001) if using MySQL
7. DBD::Pg (2.7.0) if using PostgreSQL
8. DBD::Oracle (1.19) if using Oracle
9. Digest::SHA (any)
10. Email::Send (2.04)
11. Email::MIME (1.904)
12. Template (2.22)
13. URI (1.73)

Optional Perl modules:

1. GD (1.200 for bug charting
2. Template::Plugin::GD::Image (any) for Graphical Reports
3. Chart::Lines (2.1) for bug charting
4. GD::Graph (any) for bug charting
5. GD::Text (any) for bug charting
6. XML::Twig (any) for bug import/export
7. MIME::Parser (5.406) for bug import/export
8. LWP::UserAgent (any) for Automatic Update Notifications
9. PatchReader (0.9.6) for pretty HTML view of patches
10. Net::LDAP (any) for LDAP Authentication
11. Authen::SASL (any) for SASL Authentication
12. Authen::Radius (any) for RADIUS Authentication
13. SOAP::Lite (0.712) for the web service interface
14. JSON::RPC (any) for the JSON-RPC interface
15. Test::Taint (any) for the web service interface
16. HTML::Parser (3.40) for More HTML in Product/Group Descriptions
17. HTML::Scrubber (any) for More HTML in Product/Group Descriptions
18. Email::Reply (any) for Inbound Email
19. TheSchwartz (any) for Mail Queueing
20. Daemon::Generic (any) for Mail Queueing
21. mod\_perl2 (1.999022) for mod\_perl

**Cài đặt Mail Transfer Agent (MTA)**

Bugzilla phụ thuộc vào sự sẵn có của hệ thống email để xác thực người dùng và các tác vụ khác.

**Chú ý:** Điều này không hoàn toàn đúng. Nó có thể thực hiện để hoàn toàn vô hiệu hóa việc gửi email, hoặc để các Bugzilla store email messages trong một file thay vì gửi chúng.

Trên Linux, Sendmail, Postﬁx, qmail và Exim là những ví dụ của MTA. Sendmail là Unix MTA chính thức, nhưng những MTA khác dễ cấu hình hơn và do đó nhiều người thay thế Sendmail bằng Postfix hoặc Exim.

Nếu bạn sử dụng Sendmail-compatible MTA thì nó phải đồng dạng với phiên bản 8.7 của Sendmail.

Bạn nên tham khảo hướng dẫn cho các MTA cụ thể mà bạn chọn để được hướng dẫn cài đặt chi tiết. Mỗi chương trình sẽ có các file cấu hình riêng của chúng.

**Cấu hình**

Đây là bước rất quan trọng. Bugzilla sau khi cài đặt có thể làm việc được hay không phụ thuộc vào bước này.

**Localconfig**

Bạn chạy checksetup.pl. Lần này checksetup.pl sẽ cho bạn biết tất cả các module được cài đặt và sẽ hiển thị một message về chúng và ghi chúng ra một file localconfig (trong thư mục bugzilla). File này chứa các thiết lập cho một số thông số của Bugzilla.

\bugzilla\localconfig

Mở file này bằng trình soạn thảo của bạn. Và bạn cần thay đổi 2 giá trị đó là $db\_driver và $db\_pass, tương ứng với loại db và password của user mà bạn sẽ tạo trong db của bạn. $db\_driver có thể là “mysql”, “Pg”, “Oracle”, hoặc “Sqlite”.

**Chú ý:** trong Oracle thì $db\_name là tên SID của db của bạn (ví dụ: “XE” nếu bạn sử dụng Oracle XE).

Bạn có thể thay đổi giá trị của webservergroup nếu web server của bạn không chạy trong “apache” group. Trên Debian, ví dụ, Apache chạy trong “www-data”. Nếu bạn chạy Bugzilla trên máy mà bạn không có quyền truy cập root thì bạn cần phải bỏ webservergroup trống, lờ đi những cảnh báo mà khi chạy checksetup.pl.

**Database server**

Ở đây thì chúng ta chỉ tìm hiểu về việc thiết lập đối với MySQL.

<http://dev.mysql.com/doc/refman/4.1/en/mysql-config-wizard-file-location.html>

**Cho phép file đính kèm lớn và nhiều comment**

Mặc định thì MySQL chỉ cho phép bạn chèn vào db các file có kích thước nhỏ hơn 1MB. Bugzilla sẽ kết hợp tất cả comment trên một bug, bug này có thể có kích thước lớn hơn 1MB.

Để thay đổi mặc định của MySQL, bạn cần chỉnh sửa file configuration (trên Windows bạn sửa file \mysql\bin\my.ini). Chúng tôi khuyên bạn cho phép kích thước ít nhất là 4MB bằng cách thêm “max\_allowed\_packet” vào cấu hình MySQL của bạn trong “[mysql]”.

[mysqld]

# Allow packets up to 4MB

max\_allowed\_packet = 4MB

**Cho phép từ nhỏ**

Mặc định, các từ phải ít nhất 4 ký tự. Điều này không phù hợp, gây ra rất nhiều lỗi do Bugzilla có các từ bị lỗi, bao gồm “cc”, “ftp”, “uri”. MySQL có thể được cấu hình lại để cho phù hợp. Bạn chỉnh sửa file \mysql\bin\my.ini

[mysqld]

# Allow small words in full-text indexes

ft\_min\_word\_len = 2 (chèn thêm dòng này vào)

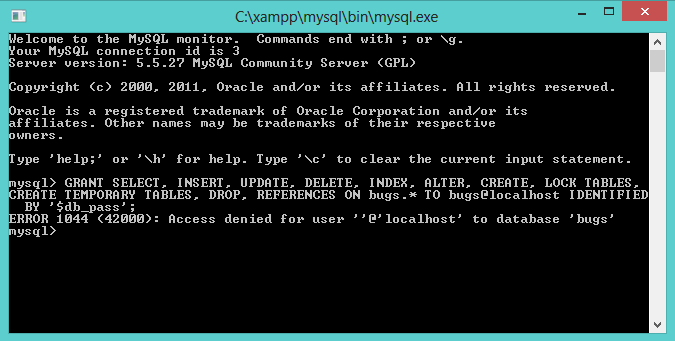
Bạn có thể tìm thêm thông tin tại <http://www.mysql.com/doc/en/Fulltext_Fine-tuning.html>

**Thêm user vào MySQL**

Bạn cần thêm một user mới để cho Bugzilla sử dụng (vì sẽ là không an toàn nếu Bugzilla sử dụng tài khoản root). Chúng ta sử dụng lệnh SQL GRANT để tạo các user. Điều này cũng hạn chế user hoạt động trong db được gọi là “bugs”, và chỉ cho phép account kết nối từ “localhost”. Modify nó để phản ánh thiết lập của bạn nếu bạn kết nối tới máy khác hoặc user khác.

Chạy mysql command-line (chạy file mysql\bin\mysql.exe)

|  |
| --- |
| **mysql>** GRAND SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, CREATE,  LOCK TABLES, CREATE TEMPORARY TABLES, DROP, REFERENCES ON bugs.\* TO bugs@localhost IDENTIFIED BY ‘$db\_pass’;  **mysql>** FLUSH PRIVILEGES; |

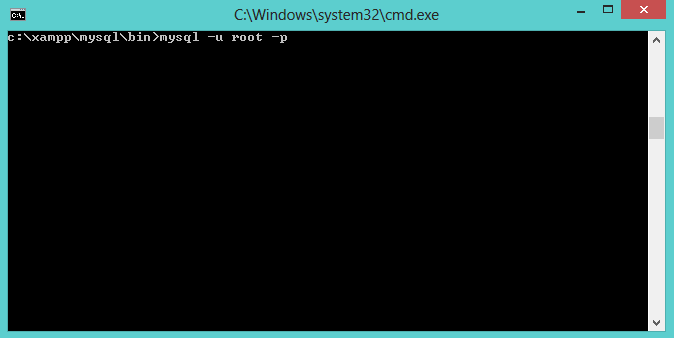


Nếu bị lỗi như trên thì bạn làm như sau:

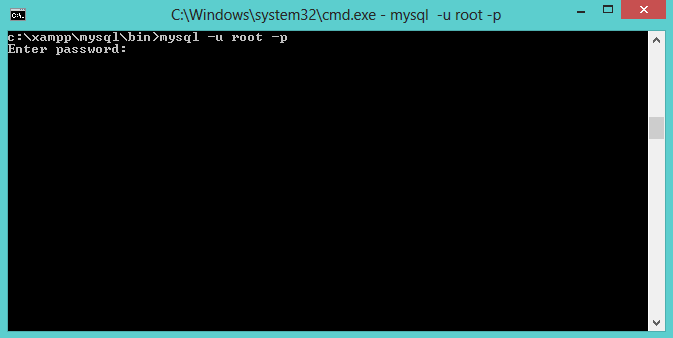
Bạn cần phải đăng nhập bằng tài khoản root.

|  |
| --- |
| > mysql -u root -p |

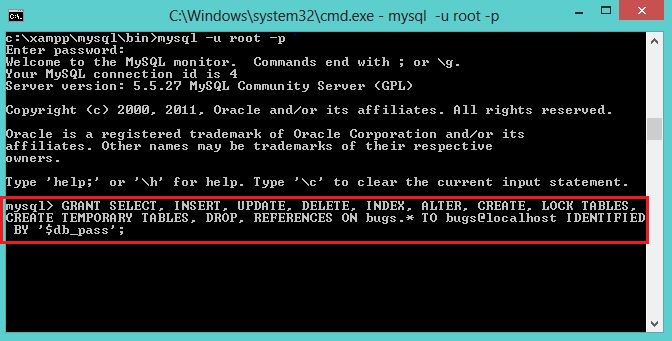
Ví dụ:



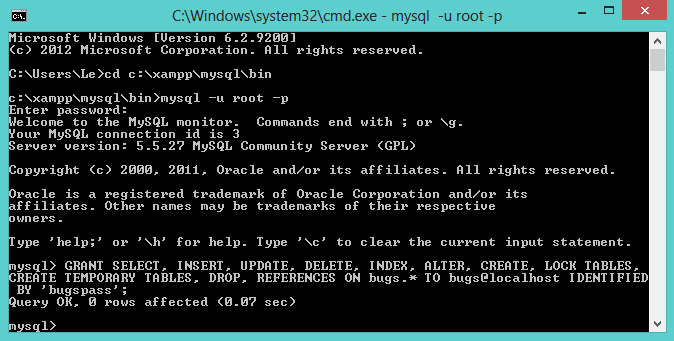
Nó sẽ yêu cầu bạn điền mật khẩu của tài khoản root.



Sau khi đăng nhập vào thành công bạn có thể viết câu lệnh GRANT.

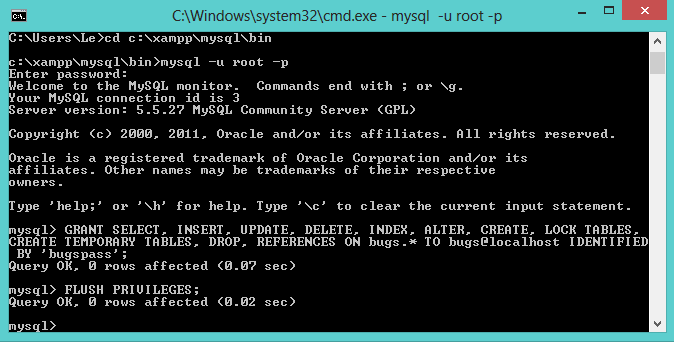


ở câu lệnh GRANT trên thì $db\_pass là mật khẩu mà bạn cấu hình trong file bugzilla\localconfig





Sau đó chạy lệnh FLUSH PRIVILEGES

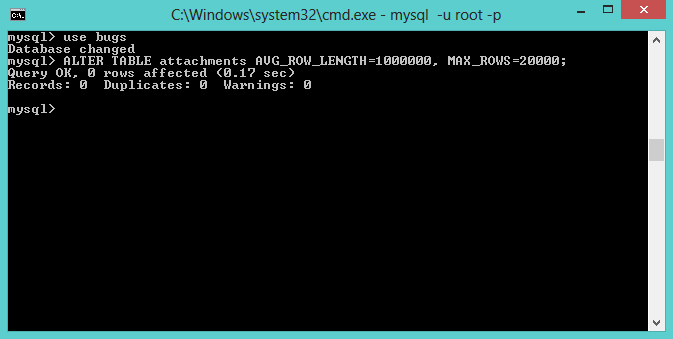


**Permit attachments table to grow beyond 4GB**

Mặc định MySQL sẽ giới hạn kích thước của một bảng là 4GB. Dưới đây là các hướng dẫn để thiết lập giới hạn cao hơn.

Sau khi bạn hoàn tất quá trình cài đặt (hoặc ít nhất là thiết lập db), bạn nên chạy MySQL commandline và nhập vào như sau (thay $bugs\_db bằng tên db Bugzilla của bạn).

|  |
| --- |
| **mysql>** use $bug\_db  **mysql>** ALTER TABLE attachments AVG\_ROW\_LENGTH=1000000, MAX\_ROWS=20000; |

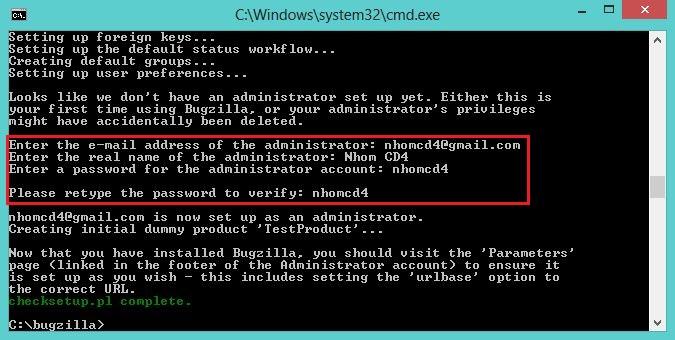


Lệnh trên sẽ thay đổi giá trị giới hạn lên 20GB. Tốt nhất bạn nên làm điều này khi bảng của bạn còn nhỏ.

**Checksetup.pl**

Tiếp theo, chạy lại checksetup.pl. Nó xác nhận lại rằng tất cả các module hiện tại và được sửa đổi trong file localconfig. Nó biên dịch các UI template, kết nối tới db sử dụng “bugs” user được tạo và password bạn định nghĩa, và tạo db “bugs” và các bảng trong db đó.

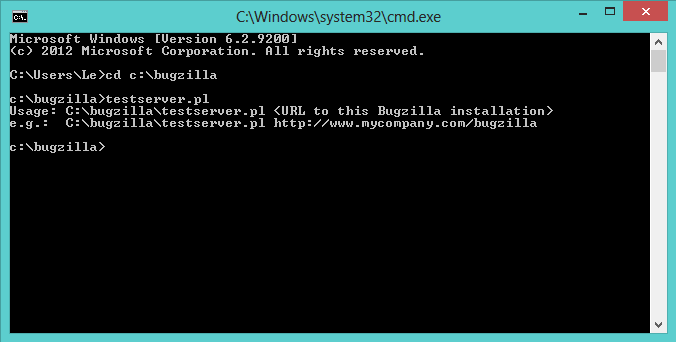
Sau đó, nó yêu cầu chi tiết về administrator account. Bugzilla có thể có nhiều administrator – bạn có thể tạo nhiều hơn một tài khoản – nhưng nó cần một tài khoản để start. Điền vào email của một administrator, tên đầy đủ, và một mật khẩu thích hợp.



**Web server**

Để kiểm tra xem web server của bạn có được cấu hình đúng hay không bạn truy cập testagent.cgi từ web server của mình. Nếu kết quả là “OK” thì cấu hình của bạn đã thành công.

Bạn có thể test server bằng bugzilla. Bạn chạy testserver.pl trong bugzilla để kiểm tra.



**Bugzilla using Apache**

Bạn có 2 lựa chọn để chạy Bugzilla dưới Apache là mod\_cgi (mặc định) và mod\_perl.

Bạn tiến hành mở file \conf\httpd.conf

Đầu tiên là thay đổi port mà Apache chạy.

|  |
| --- |
| #  # Listen: Allows you to bind Apache to specific IP addresses and/or  # ports, instead of the default. See also the <VirtualHost>  # directive.  #  # Change this to Listen on specific IP addresses as shown below to  # prevent Apache from glomming onto all bound IP addresses.  #  #Listen 0.0.0.0:80  #Listen [::]:80  **Listen 80** |

Tiếp theo là thay đổi giá trị của DocumentRoot, thay đổi giá trị của nó thành “C:/Bugzilla”.

|  |
| --- |
| #  # DocumentRoot: The directory out of which you will serve your  # documents. By default, all requests are taken from this directory, but  # symbolic links and aliases may be used to point to other locations.  #  DocumentRoot "C:/bugzilla"  #  # Each directory to which Apache has access can be configured with respect  # to which services and features are allowed and/or disabled in that  # directory (and its subdirectories).  #  # First, we configure the "default" to be a very restrictive set of  # features.  #  <Directory />  AllowOverride none  Require all denied    </Directory>  #  # Note that from this point forward you must specifically allow  # particular features to be enabled - so if something's not working as  # you might expect, make sure that you have specifically enabled it  # below.  #  #  # This should be changed to whatever you set DocumentRoot to.  #  <Directory "C:/bugzilla"> |

**Cấu hình CGI**

Tiến hành cấu hình CGI, bạn thêm dòng **AddHandler cgi-script .cgi** vào

|  |
| --- |
| #  # AddHandler allows you to map certain file extensions to "handlers":  # actions unrelated to filetype. These can be either built into the server  # or added with the Action directive (see below)  #  # To use CGI scripts outside of ScriptAliased directories:  # (You will also need to add "ExecCGI" to the "Options" directive.)  #  **AddHandler cgi-script .cgi** |

Tiếp theo chúng ta thêm tùy chọn **ExecCGI** và thay đổi **AllowOverride None** thành **AllowOverride All**. Và Apache cũng cần biết sử dụng Perl để thực thi các file .cgi thông qua chỉ thị **ScriptInterpreterSource**.

|  |
| --- |
| <Directory "C:/bugzilla">  #  # Possible values for the Options directive are "None", "All",  # or any combination of:  # Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews  #  # Note that "MultiViews" must be named \*explicitly\* --- "Options All"  # doesn't give it to you.  #  # The Options directive is both complicated and important. Please see  # http://httpd.apache.org/docs/2.2/mod/core.html#options  # for more information.  #  **Options Indexes FollowSymLinks ExecCGI**  #  # AllowOverride controls what directives may be placed in .htaccess files.  # It can be "All", "None", or any combination of the keywords:  # Options FileInfo AuthConfig Limit  #  **AllowOverride All**  #  # Controls who can get stuff from this server.  #  Order allow,deny  Allow from all    #  # Tell Apache to use Perl to execute .cgi  #  **ScriptInterpreterSource Registry-Strict**  </Directory> |

Bạn cũng nên thêm **index.cgi** vào danh sách **DirectoryIndex**.

|  |
| --- |
| #  # DirectoryIndex: sets the file that Apache will serve if a directory  # is requested.  #  # The index.html.var file (a type-map) is used to deliver content-  # negotiated documents. The MultiViews Option can be used for the  # same purpose, but it is much slower.  #  **DirectoryIndex index.html index.html.var index.cgi** |

Để **ScriptInterpreterSource Registry-Strict** có thể làm việc, bạn cần thêm một entry vào Registry để Apache sử dụng Perl để thực thi các file .cgi.

Bạn tạo một khóa **HKEY\_CLASSES\_ROOT\.cgi\Shell\ExecCGI\Command** với giá trị mặc định là đường dẫn đầy đủ của perl.exe với một parameter -T. Chẳng hạn **C:\Perl\bin\perl.exe –T**

**Disable Logging**

Để disable logging thì bạn cần comment chỉ thị **CustomLog**.

|  |
| --- |
| #  # The location and format of the access logfile (Common Logfile Format).  # If you do not define any access logfiles within a <VirtualHost>  # container, they will be logged here. Contrariwise, if you \*do\*  # define per-<VirtualHost> access logfiles, transactions will be  # logged therein and \*not\* in this file.  #  **#CustomLog logs/access.log common** |

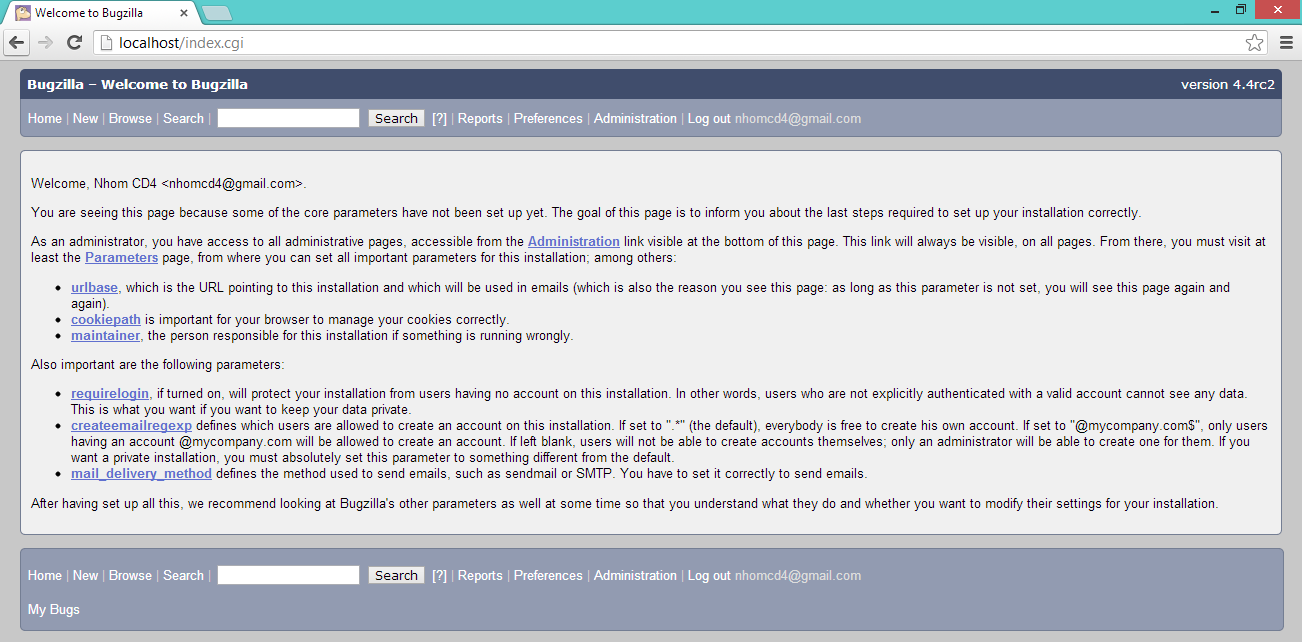
Sau khi đã hoàn tất việc cấu hình Apache bạn thực hiện restart Apache.

**Administrator**

Sau khi tiến hành cài đặt và cấu hình thành công, bạn mở trình duyệt và gõ vào “localhost”



Tại đây, bạn có thể đăng nhập vào bằng tài khoản Administrator mà bạn đã tạo trước đó (đã tạo trong quá trình cài đặt). Khi đó giao diện trang chủ sau khi bạn đã đăng nhập như sau:

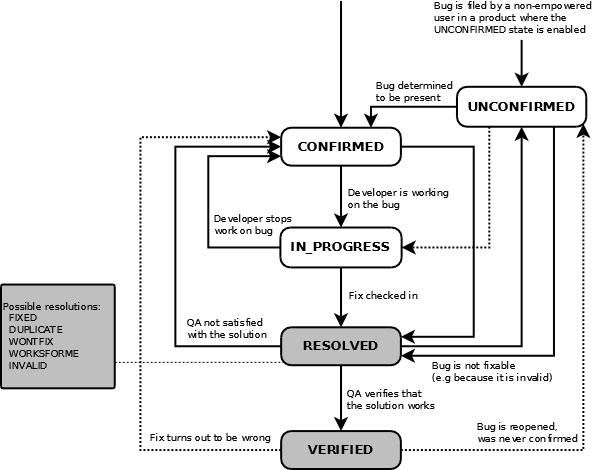


Vào vùng quản lý của Adminitrator chúng ta sẽ thấy được các chức năng chính như sau:

* Parameter : Thiết lập thông số cốt lõi của tiếng trình cài đặt.
* Default Preferences : Thiết lập các giới hạn tùy chỉnh của các User.
* Sanity Check : Kiểm thử tính thông suốt của tiến trình.
* Users : Tạo mới, chỉnh sửa, thiết lập các tài khoản user.
* Products : Chỉnh sửa thuộc tính của dự án đang test (hạn chế, thiết lập truy cập, chỉnh sửa thông tin....).
* Flags : Thiết lập cờ, tùy chỉnh các trạng thái bug và hạn chế trạng thái cho các user.
* Custom Fields : định nghĩa các fields.
* Field values: Thêm bớt, chỉnh sửa, thiết lập các giá trị của field.
* Bug Status Workflow : Tùy chỉnh công việc và thiết lập trạng thái bug ban đầu, và trạng thái cho phép chuyển tiếp lỗi khi fix.

**Vòng đời của một bug**

**Vòng đời của một bug trong Bugzilla**



Vòng đời của bug trong bugzilla 4.4rc2

**UNCONFIRMED**

Trạng thái này là người test lỗi (1 user bình thường) phát hiện và post (được cho là bug) lên hệ thống, khi ở trạng thái này, bug còn chưa được xác định nó có phải bug hay không, cần phải có người kiểm tra lại. Ở trạng thái này bug có thể chuyển sang các trạng thái như: CONFIRMED, RESOLVED, IN\_PROGRESS.

**CONFIRMED**

Bug ở trạng thái UNCONFIRMED được xác định 🡪 chuyển qua trạng thái CONFIRMED. Ở trạng thái này, bug được dev sửa lỗi. Bug có thể chuyển sang các trạng thái như: RESOLVED, IN\_PROGRESS.

**IN\_PROGRESS**

Là trạng thái tiếp theo của bug, sau khi bug CONFIRMED được fix. Nếu bug không được fix nữa, quay trở về trạng thái CONFIRMED, ngược lại thì chuyển sang RESOLVED.

**RESOLVED**

Trạng thái này bug đã được fix xong. Kết quả có thể là FIXED, DUPLICATE, WONTFIX, WORKSFORME, INVALID.

FIXED: bug đã được fix xong.

DUPLICATE: trùng với 1 bug nào đó đã tồn tại.

WONTFIX: bug này ko được fix vì lý do nào đó

WORKSFORME:

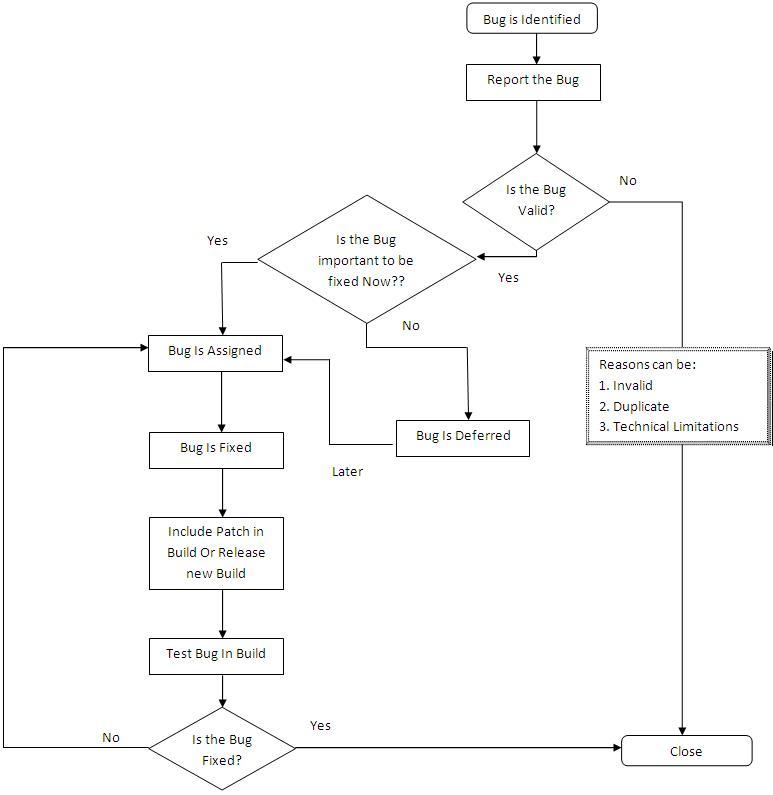
INVALID: bug không hợp lệ, lỗi không do bug

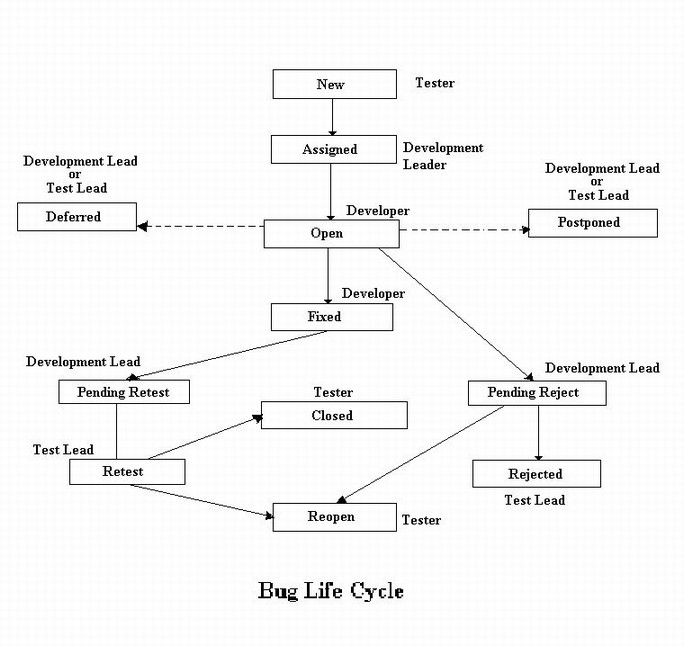
Từ trạng thái này, bug có thể chuyển qua : CONFIRMED, UNCONFIRMED hoặc VERIFIED.

**VERIFIED**

Trạng thái này do tester đã test lại xong và xác nhận đã fix bug này. Trạng thái này có thể chuyển sang trạng thái CONFIRMED hoặc UNCONFIRMED.

**Dưới đây là vòng đời khác của một bug** (có thể dùng để tham khảo)





Vòng đời của bug có thể chi nhỏ ra 7 vòng đời khác nhau:

**Cycle 1:**

Tester tìm thấy bug và báo lại cho Test Lead.

Test Leader xác nhận bug có hợp lệ hay không.

Bug không hợp lệ, bị loại bỏ.

Kết thúc.

**Cycle 2:**

Tester tìm thấy bug vào báo lại cho Test Lead.

Test Leader xác nhận bug có hợp lệ hay không.

Bug hợp lệ 🡪 được chuyển đến cho dev team. Trạng thái hiện tại của bug là “NEW”.

Dev team xác nhận lại mức hợp lệ của bug. Bug ko hợp lệ được gán trạng thái “chờ từ chối” “PENDING REJECT” trước khi trả về cho test team.

Sau khi nhận được phản hồi từ dev team, test team gán trạng thái “REJECT” cho bug.

Kết thúc.

**Cycle 3:**

Tester tìm thấy bug vào báo lại cho Test Lead.

Test Leader xác nhận bug có hợp lệ hay không.

Bug hợp lệ 🡪 được chuyển đến cho dev team. Trạng thái hiện tại của bug là “NEW”.

Dev team xác nhận lại mức hợp lệ của bug. Bug hợp lệ được gán trạng thái “ASSIGNED” và giao cho 1 dev để fix.

Dev sửa lỗi và gán trạng thái “FIXED” cho bug, sau đó trả về cho dev leader.

Dev leader gán trạng thái “PENDING RETEST” cho bug và gửi trả về cho test team để kiểm tra lại.

Test leader gán trạng thái “RETEST” cho bug và chuyển cho tester để kiểm lại.

Bug được fix thành công, được gán trạng thái “CLOSE”.

Kết thúc.

**Cycle 4:**

Tester tìm thấy bug vào báo lại cho Test Lead.

Test Leader xác nhận bug có hợp lệ hay không.

Bug hợp lệ 🡪 được chuyển đến cho dev team. Trạng thái hiện tại của bug là “NEW”.

Dev team xác nhận lại mức hợp lệ của bug. Bug hợp lệ được gán trạng thái “ASSIGNED” và giao cho 1 dev để fix.

Dev sửa lỗi và gán trạng thái “FIXED” cho bug, sau đó trả về cho dev leader.

Dev leader gán trạng thái “PENDING RETEST” cho bug và gửi trả về cho test team để kiểm tra lại.

Test leader gán trạng thái “RETEST” cho bug và chuyển cho tester để kiểm lại.

Tester kiểm lại bug, phát hiện lỗi tương tự (hoặc lỗi ban đầu) vẫn tồn tại, bug đang ở trạng thái “REOPEN” và được gửi lại cho dev team để sửa lỗi.

Kết thúc.

**Cycle 5:**

Tester tìm thấy bug vào báo lại cho Test Lead.

Test Leader xác nhận bug có hợp lệ hay không.

Bug hợp lệ 🡪 được chuyển đến cho dev team. Trạng thái hiện tại của bug là “NEW”.

Dev team cố gắng kiểm tra nếu bug hợp lệ nhưng không phát hiện lỗi tại thời điểm kiểm tra và yêu cầu hỗ trợ từ test team.

Tester cũng không tìm thấy lỗi với bug được trả về. Bug được gán trạng thái “REJECTED”.

Kết thúc.

**Cycle 6:**

Sau khi xác nhận rằng các dữ liệu là không có hoặc một số chức năng không có sẵn, bug được hoãn lại vô thời hạn và có trạng thái “POSTPONED”.

Kết thúc.

**Cycle 7:**

Nếu bug không quan trọng và cần hoãn xử lý thì có trạng thái “DEFERRED”.

Kết thúc.

Nguồn tham khảo

<http://www.buzzle.com/editorials/4-6-2005-68177.asp>