COSMO BUGZILLA tutorial

Cosmin BARBU
Massimo MILELLI

COSMO BUGZILLA: A BRIEF TUTORIAL

INDEX

WHAT IS BUGZILLA?

Bugzilla is a web-based general-purpose bug-tracker and testing tool originally developed and used by the Mozilla project, and licensed under the Mozilla public license.


Buggie, Mascot of *Bugzilla*

Released as open-source software by Netscape Communications in 1998, it has been adopted by a variety of organizations for use as a bug tracking system for both free and open-source software and proprietary projects and products. Bugzilla is used, among others, by the Mozilla Foundation, WebKit, the Linux kernel, FreeBSD, GNOME, KDE, Apache, Red Hat, Eclipse and LibreOffice.

General information

Original author(s)	Terry Weissman
Developer(s)	Mozilla Foundation
Initial release	August 26, 1998; 16 years ago
Written in	Perl
Operating system	Cross-platform
Available in	Multiple languages
Type	Bug tracking system
License	Mozilla Public License
Website	www.bugzilla.org

HOW DO I GAIN ACCESS?

The COSMO Bugzilla tracker: http://cosmo-model.org/tracker/

Fig.1: COSMO Bugzilla tracker webpage

In order to use Bugzilla you need to create an account. To create an account you first click the "**New Account**" link on the main page. Other option is to click on the "**Open a New Account**" picture. (fig.1)
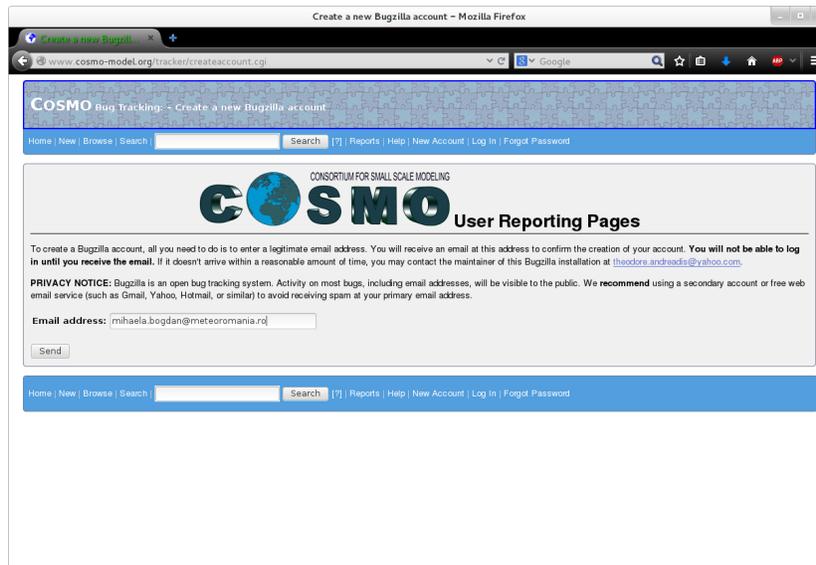
Fig.2: COSMO Bugzilla new account

Then, enter your valid email address and press "**Send**" button. (fig.2)
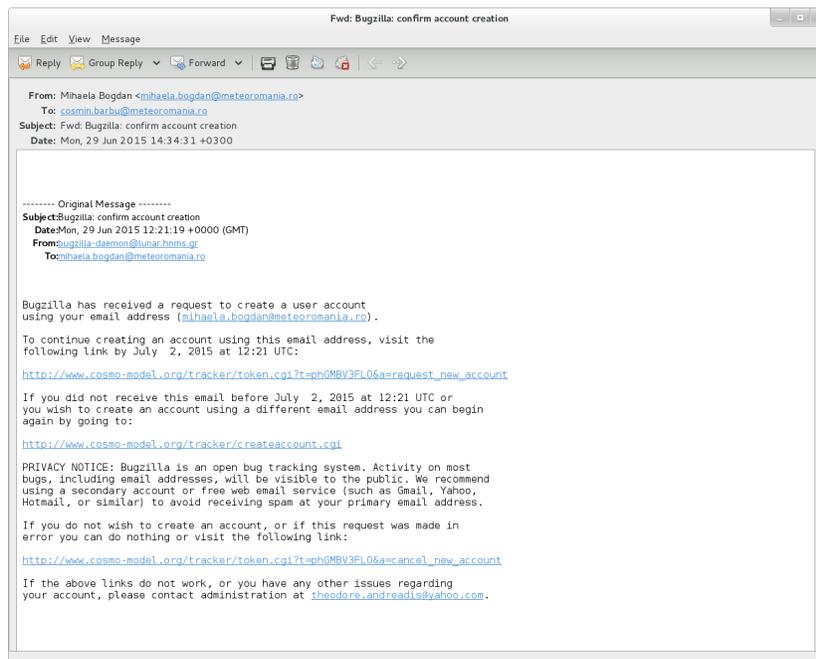
Fig.3: Email confirmation of the new account

Next you should receive an email to the address you provided, which contains a link for completing the new account creation. (fig. 3)
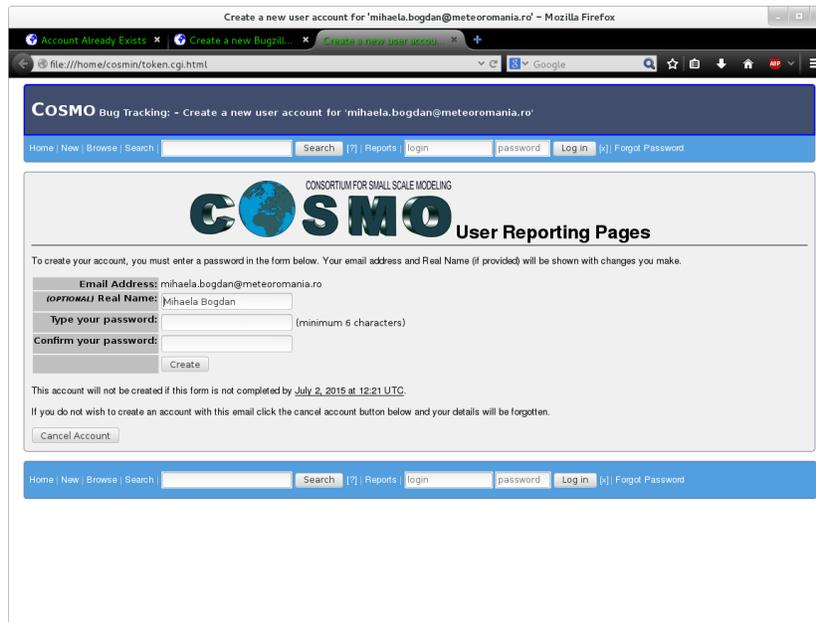
Fig.4: Last step of the new account creation

Then, the last step is to specify your login name (this is optional) and a password. Choose a strong password (memorable in the same time) and then press "**Create**" button. (fig. 4).

In this way you finished the new account creation. For logging out, it is enough to use "**Log out**" link from the page.
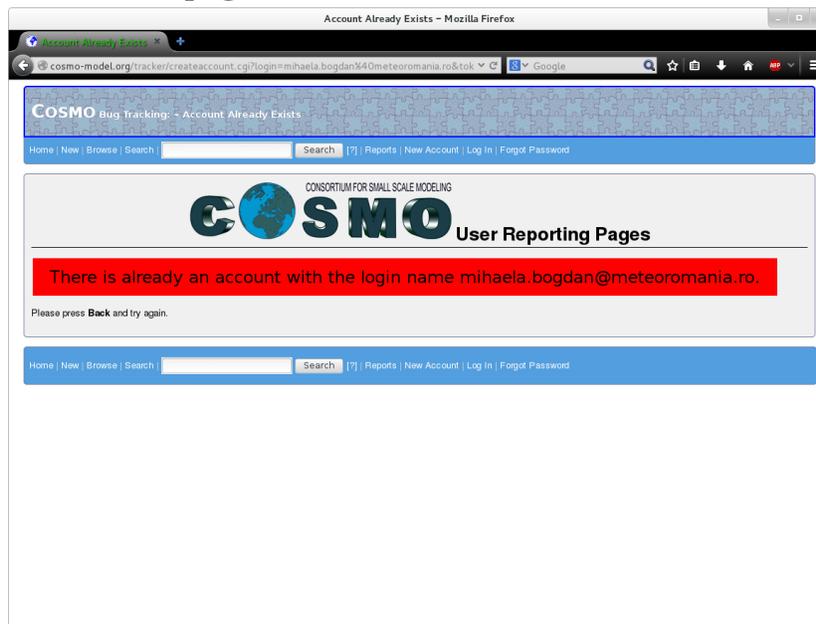

Fig.5: Example of warning message for a new account creation

In case you already have a COSMO Bugzilla account and you are trying to create a new one, you will receive a warning message (see fig. 5). But, you can simply recovery your password if you press "**Forgot Password**" link in the same warning webpage.

HOW DO I CHANGE MY ACCOUNT SETTINGS?

Once you have logged in, you can customize various aspects of Bugzilla through the "**Preferences**" link in the page footer.

The preferences are split into five tabs: "General Preferences", "Email Preferences", "Saved Searches", "Account Information" and "Permissions". (fig.6)
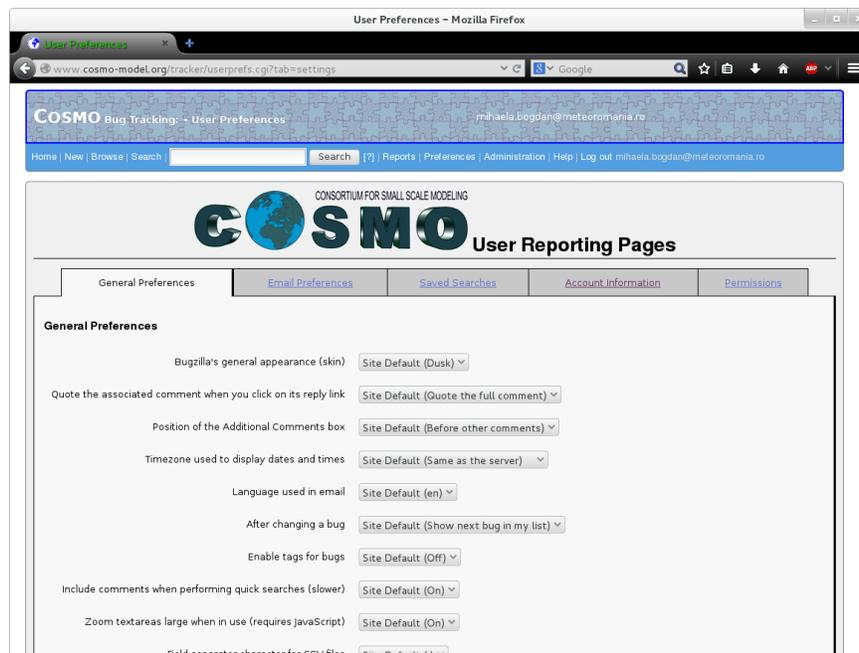
Fig.6: Account settings

"**General Preferences**": here you can find many options like chanching Bugzilla's general appearance (skin), Timezone used to display dates and times, Language used in email, Preferred email format etc etc.

"**Email Preferences**": This tab controls the amount of email Bugzilla sends you. In general, users have almost complete control over how much (or how little) email Bugzilla sends them. If you want to receive the maximum amount of email possible, click the "Enable All Mail" button. If you don't want to receive any email from Bugzilla at all, click the "Disable All Mail" button to fine-tune your bug mail, decide the events for which you want to receive bug mail; then decide if you want to receive it all the time (enable the checkbox for every column), or only when you have a certain relationship with a bug (enable the checkbox only for those columns).

"**Saved searches**": Bugzilla allows you to save searches and have them displayed in the footer of every page. On the saved searches tab you can control which searches are displayed to you, simply by un-checking the box.

"**Account Information**": On this tab, you can change your basic account information, including your password, email address and real name. For security reasons, in order to change anything on this page you must type your current password into the "Password" field at the top of the page. If you attempt to change your email address, a confirmation email is sent to both the old and new addresses, with a link to use to confirm the change.

"**Permissions**": This is a purely informative page which outlines your current permissions on this installation of Bugzilla - what product groups you are in, and whether you can edit bugs or perform various administration functions.

HOW ARE BUGS ORGANIZED?

A "bug" in Bugzilla refers to an issue entered into the database which has an associated number, assignments, comments, etc. Some also refer to a "tickets" or "issues"; in the context of Bugzilla, they are synonymous.

The broadest category of bugs is products. Normally products represent a single piece of software or entity. The products of COSMO Bugzilla are:
- Model bugs
- User Support
- web site

Model bugs:

bug reports about the main models: COSMO and Int2lm
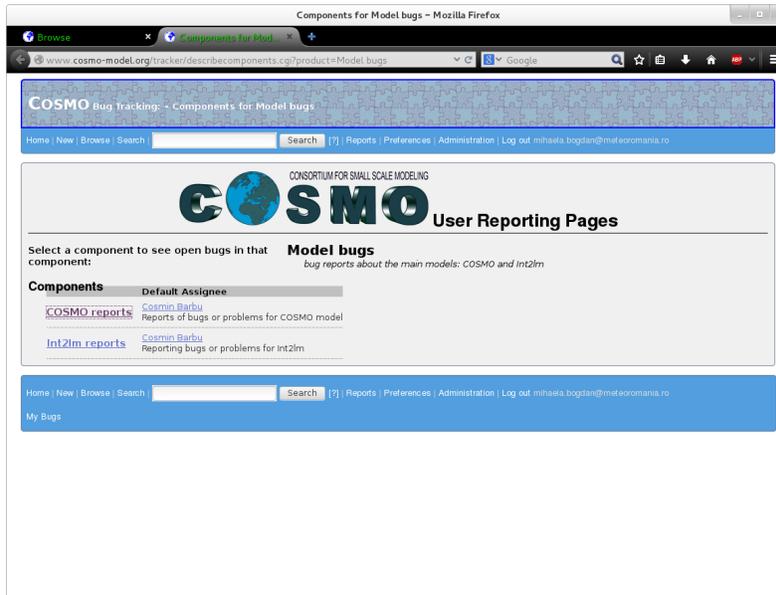
Fig.7: Model bugs reports

User Support:

support for installation and configuration of COSMO model for licensed COSMO users (default assigner is Ulrich Schaettler (DWD)) and for unlicensed COSMO users (default assigner is Cosmin Barbu (NMA)).
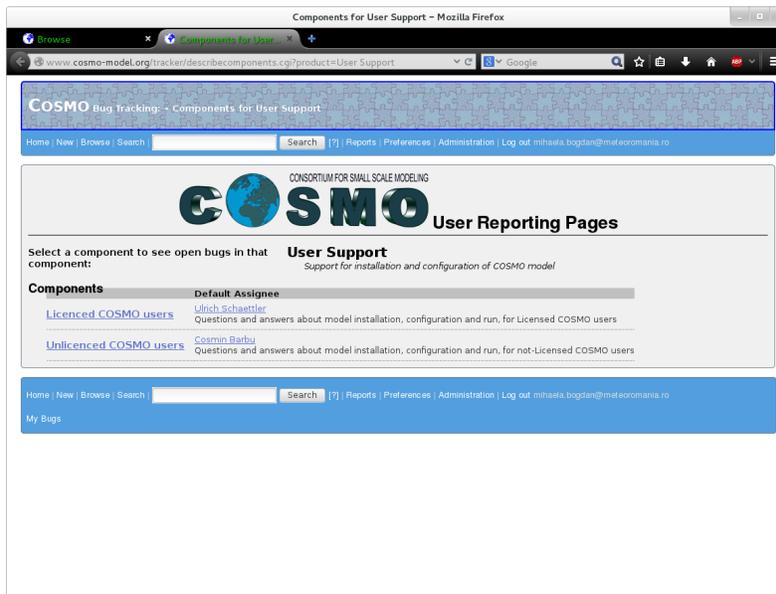
Fig.8: User support reports

Web-site:

Reporting errors or requests for the COSMO web site. The default assigner is Theodore Andreadis (HNMS).
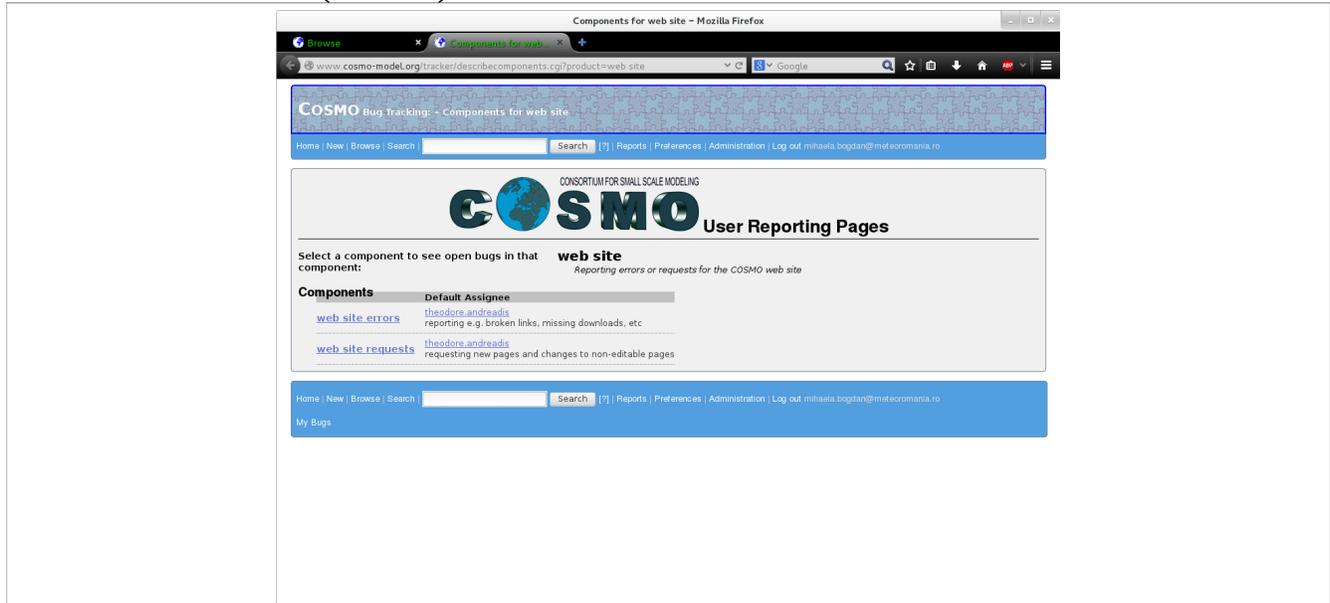
Fig.9: Web-site reports

When you are located into a Bugzilla specific product, you have four options:
- File a new bug in the "actual" product
- File a new bug in a different product
- Edit this search
- Start a new search

HOW DO I SEARCH A BUG?

Before you enter your bug, make sure if the bug has not been previously reported (already-reported bug). You can use the "Bugzilla Query Form". Here you have two options: simple or advanced search. (fig.10)

The Bugzilla Search page is the interface where you can find any bug report, comment, or patch currently in the Bugzilla system. The Search page has controls for selecting different possible values for all of the fields in a bug, as described above. For some fields, multiple values can be selected. In those cases, Bugzilla returns bugs where the content of the field matches any one of the selected values. If none is selected, then the field can take any value.
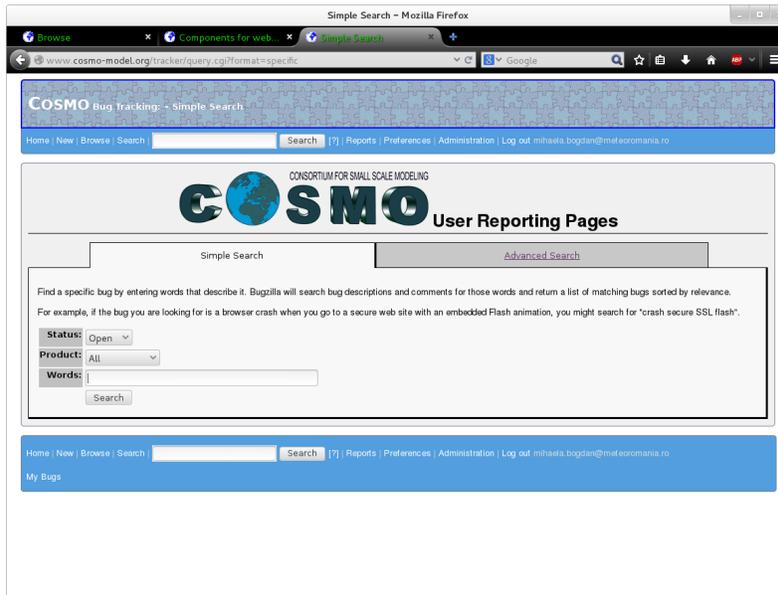
Fig.10: Bugzilla Query Form

Once you've run a search, you can save it as a Saved Search, which appears in the page footer. Once you've run a search, you can save it as a Saved Search, which appears in the page footer.
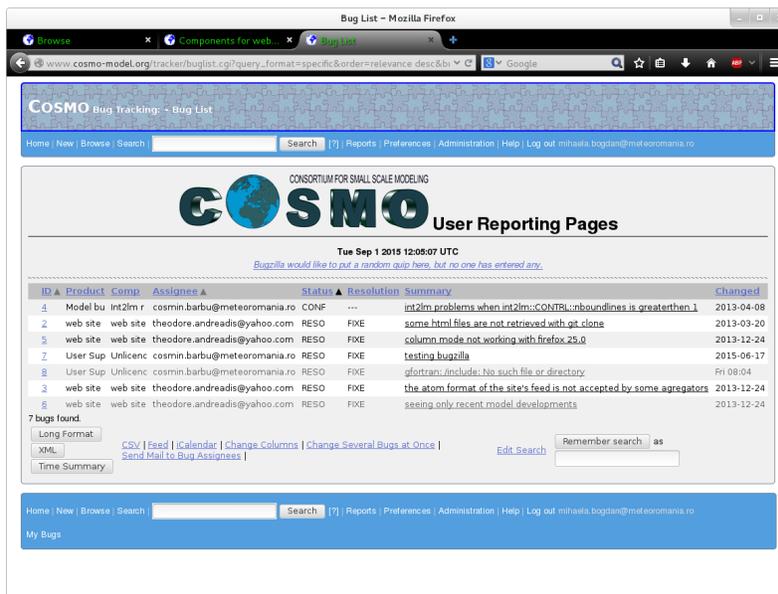

Fig.11: Example of COSMO Bugzilla searching list

When you hit search, Bugzilla will return a page with the bugs that matches your search criteria (fig. 11). The format of the list is configurable. For example, it can be sorted by clicking the column headings.
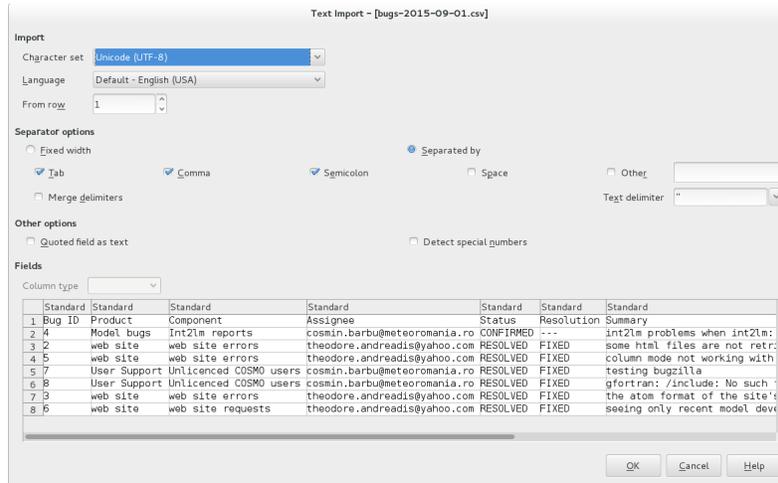
Fig.12: Example of COSMO Bugzilla searching list

Other useful features can be accessed using the links at the bottom of the list:

- **Long Format**: this gives you a large page with a non-editable summary of the fields of each bug.
- **CSV:** get the bug list as comma-separated values, for import into e.g. a spreadsheet. (fig. 12)
- **ChangeColumns**: change the bug attributes which appear in the list. Change several bugs at once: If your account is sufficiently empowered, you can make the same change to all the bugs in the list - for example, changing their owner. Send mail to bug owners: Sends mail to the owners of all bugs on the list.
- **Edit Search**: If you didn't get exactly the results you were looking for, you can return to the Query page through this link and make small revisions to the query you just made so you get more accurate results.
- **Remember Search As**: You can give a search a name and remember it; a link will appear in your page footer giving you quick access to run it again later

WHAT IS THE LIFE-CYCLE OF A BUG?

Like everything in life a bug must be born and die. **The Status field** indicates the current state of a bug. Only certain status transitions are allowed.

New bug from a
user with canconfirm
or a product without
UNCONFIRMED state

UNCONFIRMED

Bug is reopened,
was never confirmed

Bug confirmed or
receives enough votes

Developer takes
possession

NEW

Ownership
is changed

Developer takes
possession

Development is
finished with bug

Possible resolutions:
FIXED
DUPLICATE
WONTFIX
WORKSFORME
INVALID
REMIND
LATER

ASSIGNED

Development is
finished with bug

Developer takes
possession

RESOLVED

Bug is closed

Issue is
resolved

QA not satisfied
with solution

QA verifies
solution worked

REOPEN

Bug is reopened

VERIFIED

Bug is reopened

Bug is closed

CLOSED

Fig.13: The bug life cycle

Transition is from any "Open state" to the"Closed state" (fig. 13).

"Open Bugs":

- UNCONFIRMED - This bug has recently been added to the database. Nobody has confirmed that this bug is valid. Users who have the "canconfirm" permission set may confirm this bug, changing its state to CONFIRMED. Or, it may be directly resolved and marked RESOLVED.

- CONFIRMED - This bug is valid and has recently been filed. Bugs in this state become IN_PROGRESS when somebody is working on them, or become resolved and marked RESOLVED.

- IN_PROGRESS - This bug is not yet resolved, but is assigned to the proper person who is working on the bug. From here, bugs can be given to another person and become CONFIRMED, or resolved and become RESOLVED.

"Closed Bugs":

- RESOLVED - A resolution has been performed, and it is awaiting verification by QA. From here bugs are either reopened and given some open status, or are verified by QA and marked VERIFIED.

- VERIFIED - QA has looked at the bug and the resolution and agrees that the appropriate resolution has been taken. This is the final status for bugs.

The Resolution field indicates what happened to the bug. Only bugs in "Closed" state will be marked with one of the resolutions:

- FIXED - a fix for this bug is checked into the tree and tested.
- INVALID - the problem described is not a bug.
- WONTFIX - the problem described is a bug which will never be fixed.
- DUPLICATE - the problem is a duplicate of an existing bug. When a bug is marked as a DUPLICATE, you will see which bug it is a duplicate of, next to the resolution.
- WORKSFORME - all attempts at reproducing this bug were futile, and reading the code produces no clues as to why the described behavior would occur. If more information appears later, the bug can be reopened.

HOW DO I SUBMIT A BUG?

If you've discovered a new bug, you could report to Bugzilla from the main page at http://www.cosmo-model.org/tracker/ , choose "File a Bug". Select the product (Model / User Support / web site) that you've
 found a bug in, and enter your e-mail address, password, and press the "Login" button.

In the next pages we will go through an example about submitting a bug / issue related to the User Support product.

Step 1: New / User Support (fig. 14)

Fig.14: Sending a User Support bug (step 1)

Step 2: Component / Unlicensed COSMO user (in case you are a not-Licensed COSMO user) - fig. 15
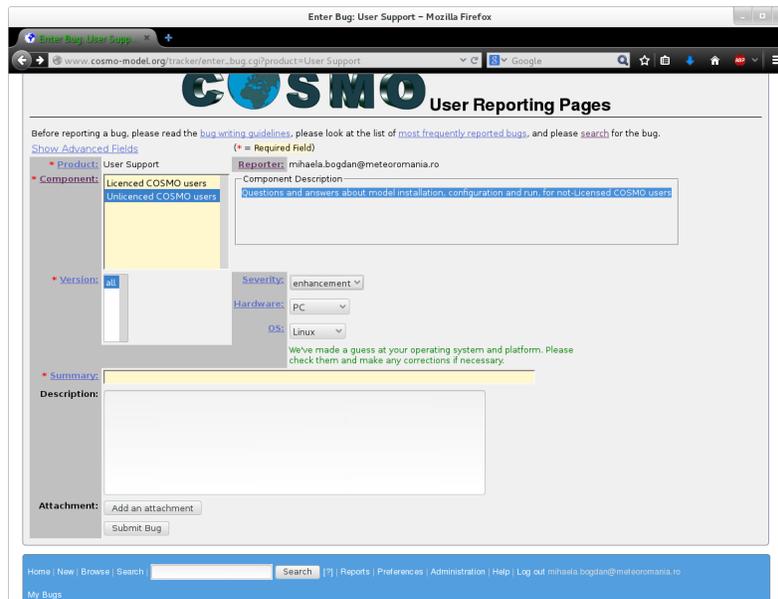
Fig.15: Sending a User Support bug (step 2)

**Step 3**: Select a field from "Severity" in order to indicate how severe the problem is. (fig. 16)
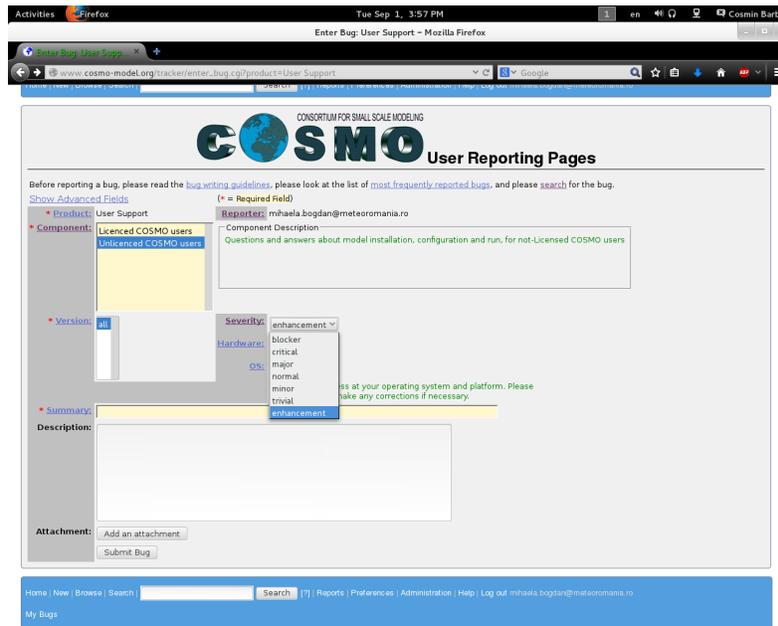
Fig.16: Sending a User Support bug (step 3)

The default value is "enhancement". To determine the most appropriate severity for a particular bug, click on the Severity link for a full explanation of each choice .

Step 4: Select a field from "Hardware" in order to indicate the hardware platform where the bug was observed on. (fig. 17)
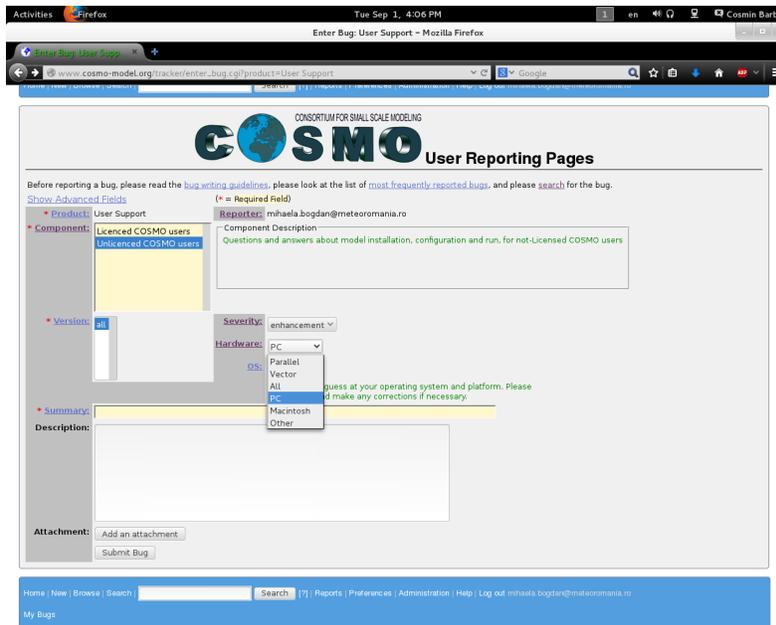
Fig.17: Sending a User Support bug (step 4)

The default value is "PC".

Step 5: Select a field from "OS" in order to indicate the operating system where the bug was observed on. (fig. 18)

Note: When searching, selecting the option "All" only finds bugs whose value for this field is literally the word "All".
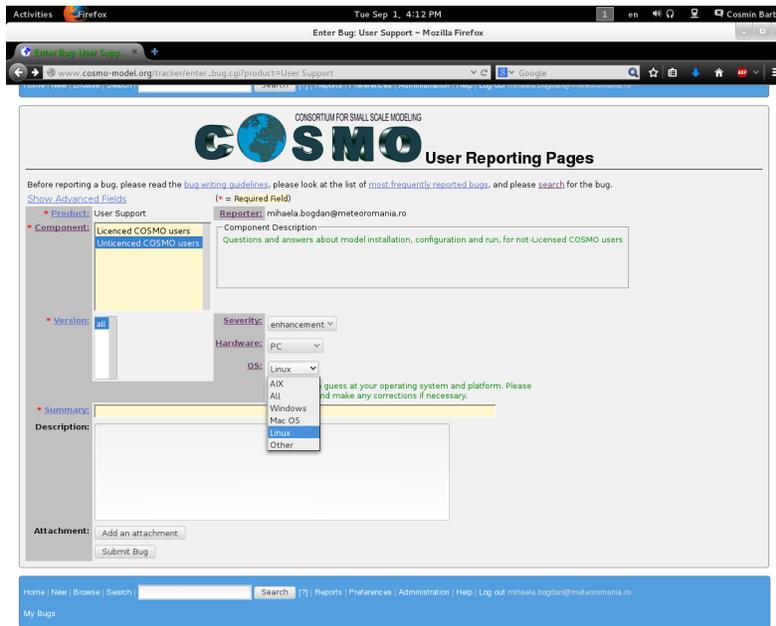
Fig.18: Sending a User Support bug (step 5)

Step 6: Send a short summary about the bug. (fig. 19)

In the summary field you must enter a short sentence which succinctly describes what the bug is about.
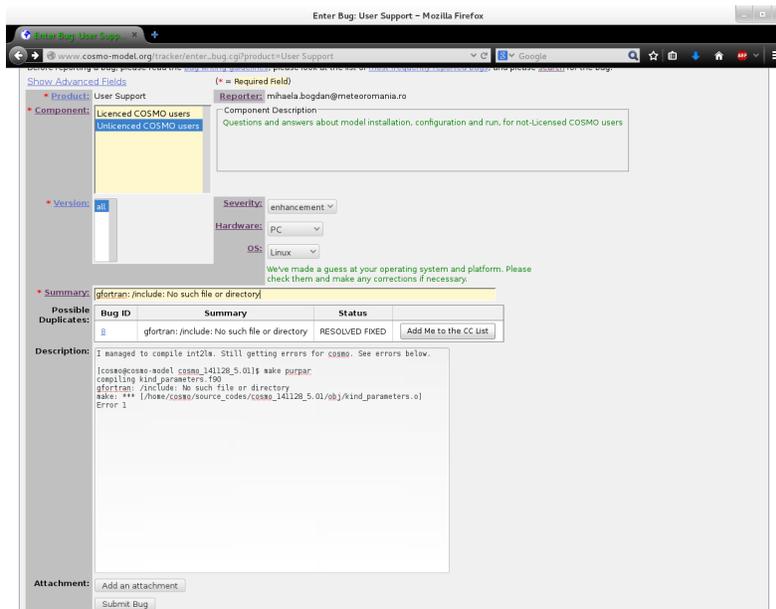
Fig.19: Sending a User Support bug (step 6)

You could also add an attachment (if you have one). In case your bug contains keywords from an already reported bug, you will be automatically warning about a possible duplicate.

Step 7: For an advanced search click on the "Show Advanced Fields".
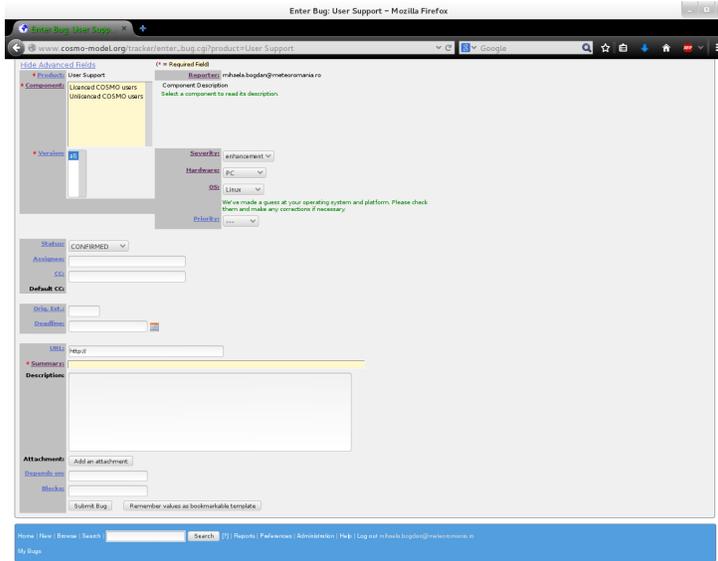
Fig.20: Sending a User Support bug (step 7)

Step 8: Send the bug by clicking on the "Submit Bug" button. Then the assignee will automatically receive an email from bugzilla-daemon@lunar.hnms.gr (fig. 21)
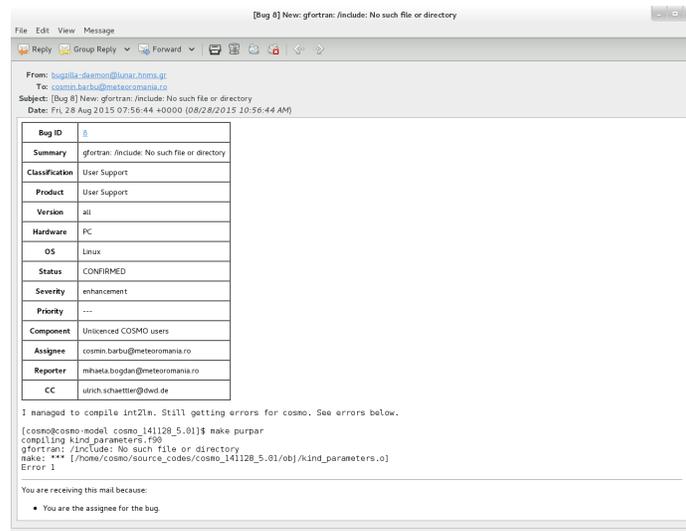
Fig.21: Sending a User Support bug (step 8)

Step 9: When the assignee will answer / solve the problem the reporter will be also automatically informed by receiving an email from <u>bugzilla-daemon@lunar.hnms.gr</u> (fig. 22).
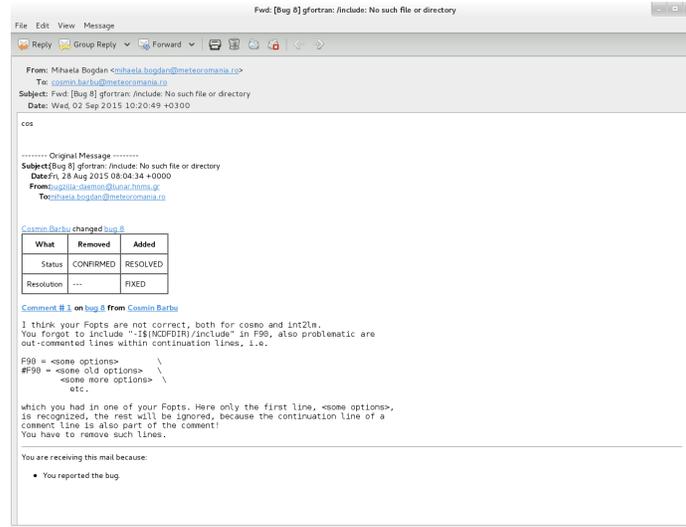
Fig.22: Sending a User Support bug (step 9)

A complete Bugzilla User guide could be found on <u>http://www.cosmo-model.org/tracker/docs/en/html/using.html</u>

BUGZILLA GLOSSARY:

A
Apache

In this context, Apache is the web server most commonly used for serving up Bugzilla pages. Contrary to popular belief, the apache web server has nothing to do with the ancient and noble Native American tribe, but instead derived its name from the fact that it was "a patchy" version of the original NCSA world-wide-web server.

B
Bug

A "bug" in Bugzilla refers to an issue entered into the database which has an associated number, assignments, comments, etc. Some also refer to a "tickets" or "issues"; in the context of Bugzilla, they are synonymous.

Bug Number

Each Bugzilla bug is assigned a number that uniquely identifies that bug. The bug associated with a bug number can be pulled up via a query, or easily from the very front page by typing the number in the "Find" box.

Bugzilla

Bugzilla is the world-leading free software bug tracking system.

C
Common Gateway Interface (CGI)

CGI is an acronym for Common Gateway Interface. This is a standard for interfacing an external application with a web server. Bugzilla is an example of a CGI application.

Component

A Component is a subsection of a Product. It should be a narrow category, tailored to your organization. All Products must contain at least one Component (and, as a matter of fact, creating a Product with no Components will create an error in Bugzilla).

Comprehensive Perl Archive Network (CPAN)

CPAN stands for the "Comprehensive Perl Archive Network". CPAN maintains a large number of extremely useful Perl modules - encapsulated chunks of code for performing a particular task.

D
daemon

A daemon is a computer program which runs in the background. In general, most daemons are started at boot time via System V init scripts, or through RC scripts on BSD-based systems. mysqld, the MySQL server, and apache, a web server, are generally run as daemons.

G
Groups

The word "Groups" has a very special meaning to Bugzilla. Bugzilla's main security mechanism comes by placing users in groups, and assigning those groups certain privileges to view bugs in particular Products in the Bugzilla database.

M
Message Transport Agent (MTA)

A Message Transport Agent is used to control the flow of email on a system. The Email::Send Perl module, which Bugzilla uses to send email, can be configured to use many different underlying implementations for actually sending the mail using the mail_delivery_method parameter. Implementations other than sendmail require that the sendmailnow param be set to on.

MySQL

MySQL is currently the required RDBMS for Bugzilla. MySQL can be downloaded from https://www.mysql.com

P
Perl Package Manager (PPM)

http://aspn.activestate.com/ASPN/Downloads/ActivePerl/PPM/

Product

A Product is a broad category of types of bugs, normally representing a single piece of software or entity. In general, there are several Components to a Product. A Product may define a group (used for security) for all bugs entered into its Components.

Perl

First written by Larry Wall, Perl is a remarkable program language. It has the benefits of the flexibility of an interpreted scripting language (such as shell script), combined with the speed and power of a compiled language, such as C. Bugzilla is maintained in

Perl.

Q
QA

"QA", "Q/A", and "Q.A." are short for "Quality Assurance". In most large software development organizations, there is a team devoted to ensuring the product meets minimum standards before shipping. This team will also generally want to track the progress of bugs over their life cycle, thus the need for the "QA Contact" field in a bug.

R
Relational DataBase Management System (RDBMS)

A relational database management system is a database system that stores information in tables that are related to each other.

Regular Expression (regexp)

A regular expression is an expression used for pattern matching.

S
Service

In Windows NT environment, a boot-time background application is referred to as a service. These are generally managed through the control panel while logged in as an account with "Administrator" level capabilities. For more information, consult your Windows manual or the MSKB.

SGML

SGML stands for "Standard Generalized Markup Language". Created in the 1980's to provide an extensible means to maintain documentation based upon content instead of presentation, SGML has withstood the test of time as a robust, powerful language. XML is the "baby brother" of SGML; any valid XML document it, by definition, a valid SGML document. The document you are reading is written and maintained in SGML, and is also valid XML if you modify the Document Type Definition.

T
Target Milestone

Target Milestones are Product goals. They are configurable on a per-Product basis. Most software development houses have a concept of "milestones" where the people funding a project expect certain functionality on certain dates. Bugzilla facilitates meeting these milestones by giving you the ability to declare by which milestone a bug

will be fixed, or an enhancement will be implemented.

Tool Command Language (TCL)

TCL is an open source scripting language available for Windows, Macintosh, and Unix based systems. Bugzilla 1.0 was written in TCL but never released. The first release of Bugzilla was 2.0, which was when it was ported to perl.

Z
Zarro Boogs Found

This is just a goofy way of saying that there were no bugs found matching your query.