



1. INTRODUCTION

bug then returns to whomever opened it to verify the solution. If the solution is acceptable, the bug can then be changed to the “closed” state. If not, it is put back into the “open” state and a developer will continue to work on it.

The distinction between the Resolved state and the Closed state is very important for quality. Some problem tracking systems (like Request Tracker) only operate in two states: Open and Closed. This does not enforce a verification step into the process, since the originator of the problem does not have the final say in whether an issue is closed or not.

1.3. Document Overview

This document provides an overview of the WebVarmint system. Section 2 provides an overview ofug describes all withugn

2. OVERVIEW OF THE BUG RECORD

Table 1. lists the fields recorded with each bug. Each field is a field in the database, and all fields together make up the record. Some of the fields may be empty. The key to each record will be the DataBase name and BugNumber. The BugNumber is a unique consecutive integer automatically generated (by the client program) when a new bug is opened.

The Description field in the bug record is the textual description typed in whenever a bug is opened, modified, assigned, resolved, or closed. Existing text can NOT be modified, you can only append additional text. When any of these operations are performed, a delimiter line is inserted into the text like this:

```
### <CurrentDate> ### Item opened by <CurrentUser> ###
### <CurrentDate> ### Item modified by <CurrentUser> ###
### <CurrentDate> ### Item assigned by <CurrentUser>: <Old> -> <New> ###
### <CurrentDate> ### Item resolved by <CurrentUser> as 2 resolutions ###
### <CurrentDate> ### Item resolved by <CurrentUser> as 2 resolutions ###
```


Query New Bug

id (bug_list.db=confia.cdef and confia. select bug_list*: onfiq where (bug_list.Status != 'closed') ar and (bug_list.db= Execute

Statistics ▶

Members ▶

Other ▶

All Current Bugs

All Current Bugs - Cur

2000...	2001...	O	250	ifc	H	S
2000...	2001...	R	252	ifc	H	S
2000...	2001...	O	261	ifc	M	S
2000...	2001...	O	267	ifc	M	S
2000...	2001...	O	271	ifc	M	S
2001...	2001...	R	286	ifc	M	S
2001...	2001...	O	290	ifc	M	S
2001...	2001...	R	310	ifc	M	S
2001...	2001...	O	315	ifc	M	S
2001...	2001...	O	334	ifc	M	S

alpho... Write a BiGraph class for bi-directional graph arcs.

duncan Finalize the design of the Graph class.

alpho... diagnose methods in Graph classes

buang update examples to use C++ and line

parihar DebugLevel class API

new... multivariate gaussian random

4.5. How to Close a Bug

Once you are viewing a bug (see Sectiona pressa Close button. You may only close a resolved bug (see the lifecycle description on page 2).

The TestCase and Description fields will now be enabled and allow user edits. The Status will be set to Closed, and the owner field will be unchanged.

The appropriate delimiter string will be inserted into the textual description,

```
### <CurrentDate> ### Item closed by <CurrentUser> ###
```

The user completes the operation by clicking Save. No fields must be entered for a bug to closed.

4.6. How to Assign a Bug

The only way to change the owner of a bug is through the Assign operation. Begin by viewing the

5. ADMINISTRATIVE FUNCTIONS

The following section describes administrative functions that sometimes need to be performed in WebVarmint. First we discuss configuration, then we discuss how to forcibly unlock all bugs.

5.1. How to Modify the System Configuration

Administrators can modify the system configuration through the Configuration Screen. This screen is accessed from Edit->Configuration on the main menu. All editable system parameters

srom the menu. mwil eb
asimply mdify the Siste as escirdttton

