|  |  |
| --- | --- |
| DEP Logo | **Department of Environmental Protection** |
| ***STD–09061804.1.0*** | ***Page 1 of 2*** |
| **Java Enterprise Edition (JEE) Testing Standard** | |

# Purpose

This document specifies the Florida Department of Environmental Protection’s (DEP) Java Enterprise Edition (JEE) Testing Standard.

# Scope

This document applies to all JEE code developed by and/or for the DEP.

# Standard

1. Each project’s test plan shall include the core standards of testing according to **Appendix A: *DEP JEE Testing Guidelines***.
2. All applications shall be tested in accordance with an approved test plan.
3. All new applications and maintenance releases shall have documented test results.
4. For User Acceptance Testing, all defects, analysis and resolution shall be tracked using DEP’s JIRA Application. Refer to **Appendix B:** ***JIRA Setup Documentation*** and **Appendix C:** ***JIRA Instructions for Logging Issues*** for details.

# Deviation from Use

Any deviation from this standard shall be documented in associated project and contract documentation. For contracts, deviation from standard shall be documented and approved by the DEP contract manager. For non-contract work, deviation from use shall be documented in the project plan/scope of work and approved by the project manager.

# Appendices

Appendix A: *JEE Testing Guidelines.*

Appendix B: *JIRA Setup Documentation.*

Appendix C: *JIRA Instructions for Logging Issues.*

**Approved by R. John Willmott, CIO 6/18/09**

Approval Date

**Appendix A: DEP JEE Testing Guidelines**

# Introduction to Testing Guidelines

This document specifies the Java Enterprise Edition (JEE) Testing Guidelines at the Department of Environmental Protection (DEP). In order to ensure delivery of quality software products, DEP requires (at a minimum) the following levels of testing.

# Phases of Testing – Development

### Unit

The initial phase of testing is the development and execution of Unit Tests on individual callable methods and behaviors of the business objects which are reflected in the Java code of the application. These tests are written using the JUnit testing framework.

### Integration

The next phase of testing is Integration (although that term has multiple meanings). For our work, this refers to testing multiple units of code to ensure that data and business logic flow in the correct order and that known input produces the correct desired output. This may be done via the usage of a “test harness” such as the Struts Test Case for Struts based applications, JMeter, or other tools which provide the requisite level of assurance that the code is operating as intended.

### Thread

The next phase of testing is Thread testing. Following data from raw input to the final output report. Ensuring that all business logic and data manipulation of a client’s unit of work flow smoothly through the system and result in the correct desired output.

### System

The final phase of development testing is the full system level. This will test all threads, and all associated metrics and reporting aspects of the system. This may be done manually or via a standard system based testing tool such as WinRunner or LoadRunner.

# Phases of Testing – Post-development

### User Acceptance Testing

The main required phase of testing post-development is that of User Acceptance Testing. This is to be done according to a documented and approved UAT Test Plan, and its results must be reported in order for the program to be approved to move into production.

# Tools

## JUnit

JUnit is a simple framework for writing and running automated tests. It may be obtained from <http://www.junit.org/>

## JMeter

Apache JMeteris a 100% pure Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions. It may be obtained from the Apache Jakarta Project library at <http://jakarta.apache.org/jmeter/>

## StrutsTestCase

StrutsTestCase for JUnit is an extension of the standard JUnit TestCase class that provides facilities for testing code based on the Struts framework. StrutsTestCase provides both a Mock Object approach and a Cactus approach to actually run the Struts ActionServlet, allowing you to test your Struts code with or without a running servlet engine. Because StrutsTestCase uses the ActionServlet controller to test your code, you can test not only the implementation of your Action objects, but also your mappings, form beans, and forwards declarations. And because StrutsTestCase already provides validation methods, it's quick and easy to write unit test cases.

It may be obtained from <http://strutstestcase.sourceforge.net/>

**Appendix B: JIRA Setup Documentation**

# Introduction

JIRA[[1]](#footnote-1) is a bug- and issue-tracking system developed by Atlassian. The Florida Department of Environmental Protection (DEP) uses JIRA to track reported issues identified during application development testing. Before JIRA can be used for issue tracking for an application, the project manager must request setup of the JIRA environment. The following sections outline the request process for a new JIRA project.

# Outline of JIRA Setup

The OTIS JIRA Administrator performs the initial setup for all JIRA projects using information provided by the application project manager. This information includes:

1. Basic descriptive information
2. Users
3. Groups
4. Project Lead
5. Permission Scheme
6. Notification Scheme

# JIRA Support

To request JIRA setup for your project, complete and email the setup documentation forms provided in the following sections to the JIRA administrator using the contact information found on the JIRA logon screen at the following URL:

<http://jhs.dep.state.fl.us:8080>

# JIRA Project and Permission Scheme Setup

Complete the **Supplementary Material/Information – Project Setup and Permissions Scheme Form** below.This form identifies what the general user will see and be able to do when they log onto JIRA. The first section of the form identifies basic application information:

|  |  |  |
| --- | --- | --- |
| **Title** | **Description** | **Example** |
| Project Category | JIRA provides a grouping function that allows projects to be linked to an overarching category for easier navigation. The Project Category identifies the name of the section in JIRA in which this project will be displayed. You may request a new category to link your new JIRA project to, identify an existing category to link your project to, or keep your project unlinked. | Database Server Upgrades  DARM Application Testing |
| Project Name | The name of the project displayed to the general user. | DBDEV Testing  ARMS Maintenance |
| Project Short Name | The acronym of the project. This acronym is used to build the directory structure and URL for your project. | DBDEV  ARMSM |
| URL | You may define a URL that can be used to directly access your JIRA project. |  |
| DEP Project Lead | The DEP Project Lead maintains the JIRA project and performs the administrative functions of the project such as adding and removing project users, managing issue resolution, and closing issues. This must be a person, not a group, and is a registered JIRA user; enter the JIRA username and full name. | gorton\_d Donna Gorton |
| Default Assignee | The Default Assignee is the JIRA user that is initially assigned a new issue when it is created. You may identify any JIRA user or leave this empty. If you leave it empty the Project Lead will have to process all issues. This must be a person, not a group, and is a registered JIRA user; enter the JIRA username and full name. | gorton\_d Donna Gorton |
| Description | The full description of the new JIRA project. You may copy the information from the Project Management documentation. | “The DBDEV Testing project was established to track all issues related to the testing of the new DEP development database server prior to the server becoming “Live”.” |

The second section of the form identifies your project users who will need JIRA user accounts but do not yet have one. Enter the DEP Network Account, Full User Name, and the User’s email address.

The third section of the form identifies new JIRA groups that are to be set up for your project. Enter the short name of the group, a brief description, and the network user account of the users who belong to the new group.

The last section of the form identifies the user and/or groups who will be given specific permissions for your project. You may list the role (Project Lead), group name (JIRA-users Group), or specific JIRA user names for each permission. The form lists the standard Permission Scheme setup used at DEP. You may leave the Permission Scheme in the standard format if you wish.

# Supplementary Material/Information – Project Setup and Permissions Scheme Form

**Project Category: [Enter the JIRA Project Category Name]**

**Project Name: [Enter the JIRA Project Name]**

**Project Short Name: [Enter the Project Short Name]**

**URL: [Enter the URL used to reach the Project – optional]**

**Project Lead: [Enter the JIRA username of the person responsible for this project]**

**Default Assignee: [Enter the JIRA username of the person who will be the Default Assignee for issues for this project]**

**Description: [Enter the description of the new project]**

| **New JIRA User**  **Network Account** | **Full User Name** | **Email Address** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

| **New Group Name** | **Description** | **Assigned Users** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

| **Permission Scheme** | |
| --- | --- |
| **Permission** | **User/Groups** |
| Administer Projects | * Project Lead |
| Browse Projects | * Project Lead * JIRA-users Group |
| Create Issues | * Project Lead * JIRA-users Group |
| Edit Issues | * Project Lead |
| Schedule Issues | * Project Lead |
| Move Issues | * Project Lead |
| Assign Issues | * Project Lead |
| Assignable User | * Project Lead * JIRA-users Group |
| Resolve Issues | * Project Lead * JIRA-users Group |
| Close Issues | * Project Lead |
| Modify Reporter | * Project Lead |
| Add Comments | * Project Lead * JIRA-users Group |
| Edit All Comments | * Project Lead |
| Edit Own Comments | * Project Lead * JIRA-users Group |
| Delete Issues | * Project Lead |
| Work On Issues | * Project Lead * JIRA-users Group |
| Link Issues | * Project Lead |
| Create Attachments | * Project Lead * JIRA-users Group |
| View Version Control | * Project Lead * JIRA-users Group |
| View Voters and Watchers | * Project Lead |
| Manage Watcher List | * Project Lead |
| Set Issue Security | * Project Lead |

# JIRA Project Notification Scheme Setup

If you choose to enable the email notification functionality for your project, complete and submit the following **Project Notification Scheme Form**. This form defines the events that will trigger an email notification to the specific roles, users, and/or groups. You may list the role (Project Lead), group name (JIRA-users Group), or specific JIRA user names for each event. The document lists the standard Notifications Scheme setup used at DEP; you may leave the Notifications Scheme in the standard format if you wish.

# Project Notification Scheme Form

**Project Name: [Enter the JIRA Project Name]**

| **Event** | **To be notified** |
| --- | --- |
| Issue Created: | * Project Lead * All Watchers |
| Issue Updated: | * All Watchers |
| Issue Assigned: | * Current Assignee * All Watchers |
| Issue Resolved: | * Project Lead * All Watchers |
| Issue Closed: | * Reporter * All Watchers |
| Issue Commented: | * All Watchers |
| Issue Reopened: | * Current Assignee * All Watchers |
| Issue Deleted: | * Project Lead * Current Assignee * All Watchers |
| Issue Moved: | * All Watchers |
| Work Logged On Issue: | * All Watchers |
| Work Started On Issue: | * Project Lead * All Watchers |
| Work Stopped On Issue: | * Project Lead * All Watchers |
| Generic Event: | * Project Lead |

**Appendix C: JIRA Instructions for Logging Issues**

**INSTRUCTIONS FOR USING THIS TEMPLATE**

|  |
| --- |
| *All Help Text in this document is in the shaded blue boxes. Delete all Help Text from your final draft! Items that are intended to stay in as part of your document are in* ***bold*** *and highlighted in yellow. Plain text is used where you might insert wording about your project.* |

Use this template to create a JIRA Instructions Guide specific to your project. Tailor/modify this document to your needs, removing explanatory comments, bold and yellow highlighting and JIRA functionality not in use by your project. You may leave the sample screenshots below but it is recommended that you replace them with screenshots of your own project to prevent confusion.

# Introduction

|  |
| --- |
| *The following subsections of this document should provide an overview of how to use JIRA for your project. You need not type any text under the INTRODUCTION section; however, all of the other sections should adequately provide instructions for selected features in use.* |

JIRA[[2]](#footnote-2) is a bug- and issue-tracking system developed by Atlassian. The Florida Department of Environmental Protection uses JIRA to track reported issues identified during application development testing. Complete and use for testing for your specific project.

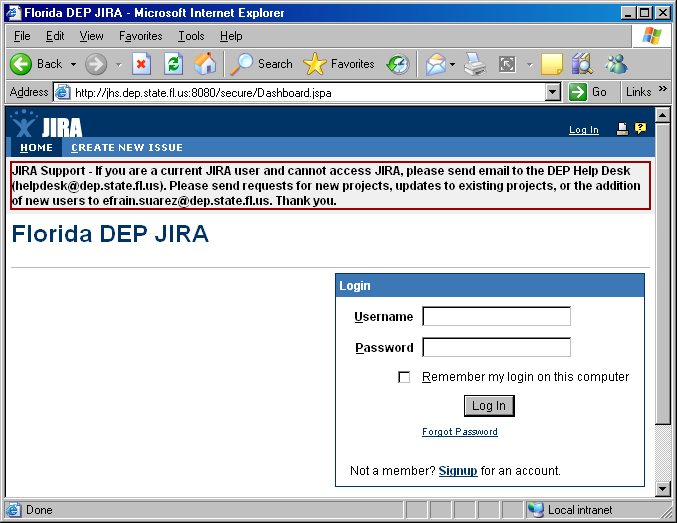
# Logging onto JIRA

You must have a JIRA account to use the JIRA application; if you do not yet have an account, contact the JIRA administrator using the contact information on the JIRA logon screen.

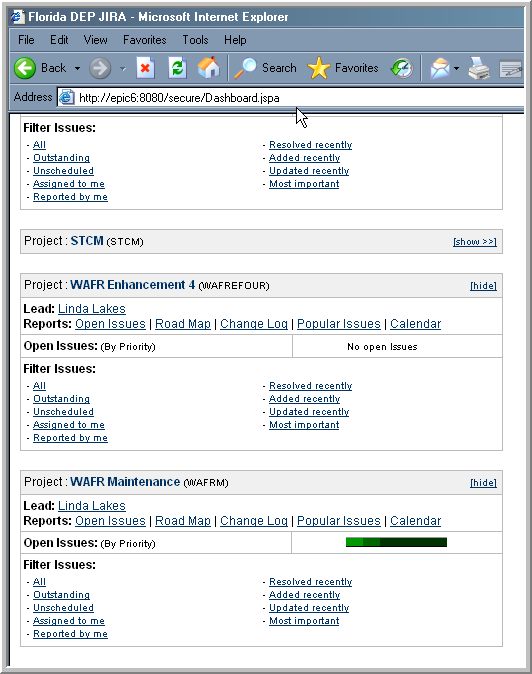
To log onto JIRA, use MS Internet Explorer and type in the following URL:

http://jhs.dep.state.fl.us:8080

A logon window will be displayed. Type in the username and password provided via email and select the *Log In* button. Passwords are case-sensitive.

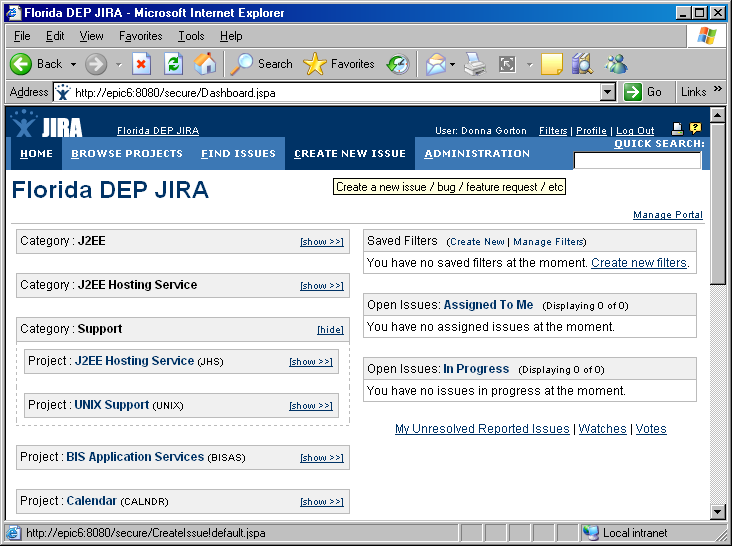


Once logged in, you will see a dashboard of all categories/projects that you can access. Use the **[your JIRA project name]** project to log problems with the **[System name and acronym]** application.



# Logging an Issue

To log an issue discovered during testing, select the *Create New Issue* menu option at the top of the screen.



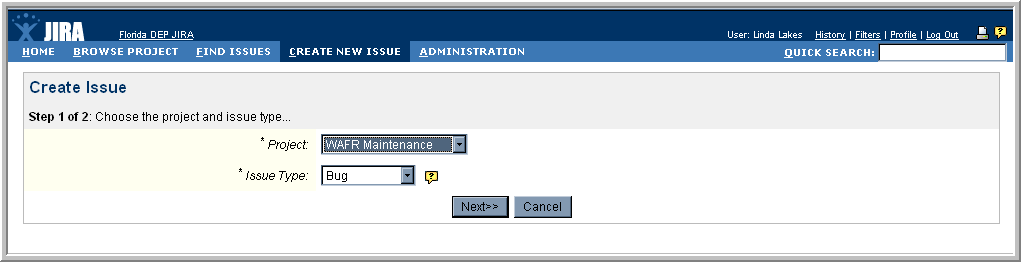
Choose **[your JIRA project name]** from the *Project* pick list.

Select an *Issue Type.* The following issue types are applicable to the **[your JIRA project name]** project:

|  |
| --- |
| *Identify the Issue Types used in this project. The JIRA Administrator can assist you with the available options.* |

* **Bug** - A problem that impairs or prevents the functions of the product.
* **Improvement** -An improvement or enhancement to an existing feature or task.
* **New Feature** -A new feature of the product, which has yet to be developed.
* **Task** -A task that needs to be completed.

Select *Next*.



*Identify the Issue Types used in this project. The JIRA Administrator can assist you with the available options.*

|  |
| --- |
| *If you do not create Components for your project, remove the Components section below.*  *If you set your permission scheme so that:*   1. *The “reporter” is not allowed to choose the Assignee, remove the Assign To section below.* 2. *The “reporter” is not allowed to choose another Reporter Name, remove the Reporter section below.* |

Fill in the details of the issue that is being reported in the following format. Select the *Create* button when finished.

**Summary**: Enter the form title in the summary field followed by a hyphen. Follow the hyphen by a summary of the issue being reported.

**Priority**: Choose the priority of this issue.

High - Major issue that must be resolved.

Medium - Moderate issue, affects primary functionality but may not block continuation of the project.

Low - Minor issue, does not cause disruption of functionality.

**Due Date**: Leave this blank.

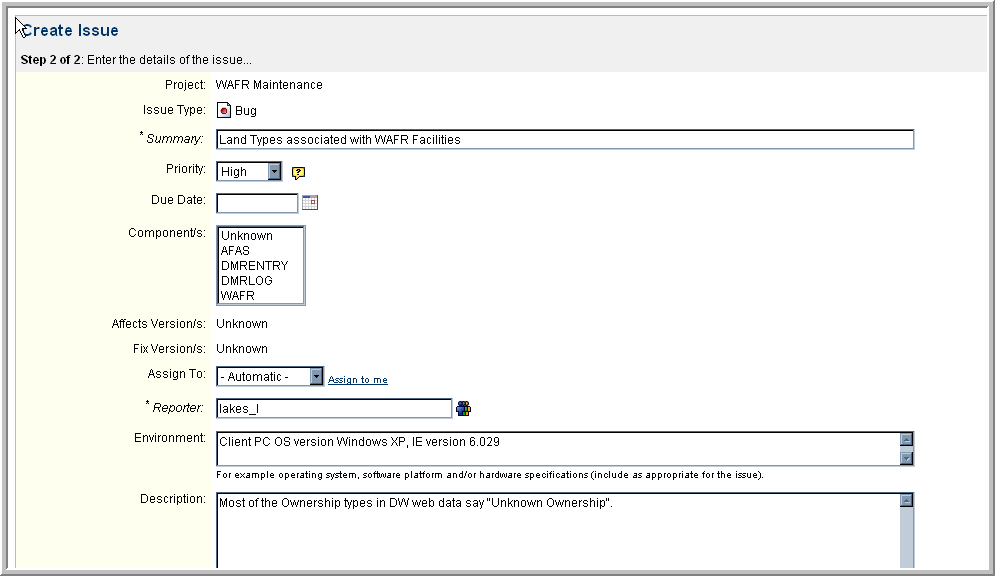
**Components**: Choose the **[your application name]** application section that you are testing.

**Assign To**: Leave this at Automatic.

**Reporter**: Leave this as your name or choose a name by selecting the *Select a User* icon.

**Environment**: Fill in information about the PC being used to test the application. Include information such as the PC operating system version and the IE browser version.

**Description**: Provide a complete description of the issue being reported. Include details that can be used to recreate the issue such as the screen on which the issue occurred, keystrokes used, data that was entered, etc.

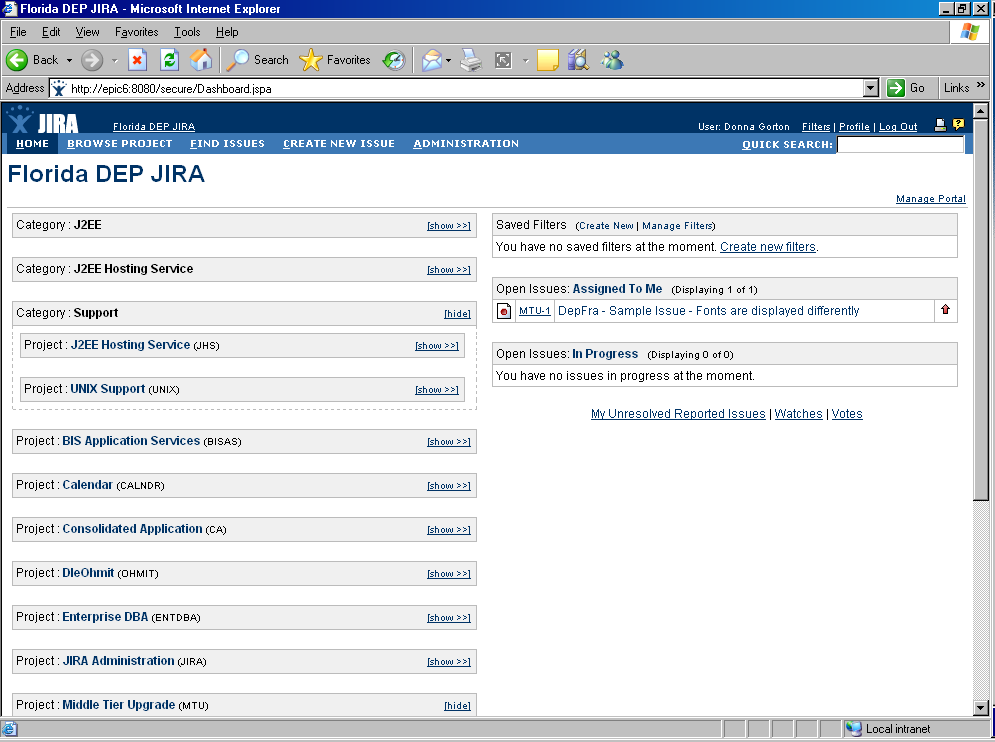


|  |
| --- |
| *Remove all items from the below paragraph that users are not allowed to perform for your project.* |

JIRA will next display the Issue Details screen listing the status of the issue as Open. On the left side of the screen are choices of operations that can be performed for the issue. Note that you may update the issue, attach screenshots and files, add comments or set the issue so that you are notified when an operation is performed on the issue.



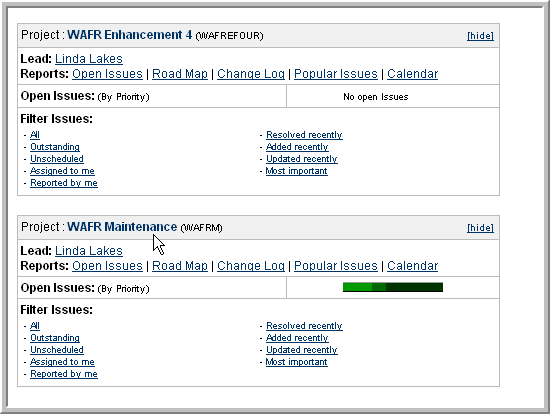
To return to the JIRA home screen, click the *Home* menu option



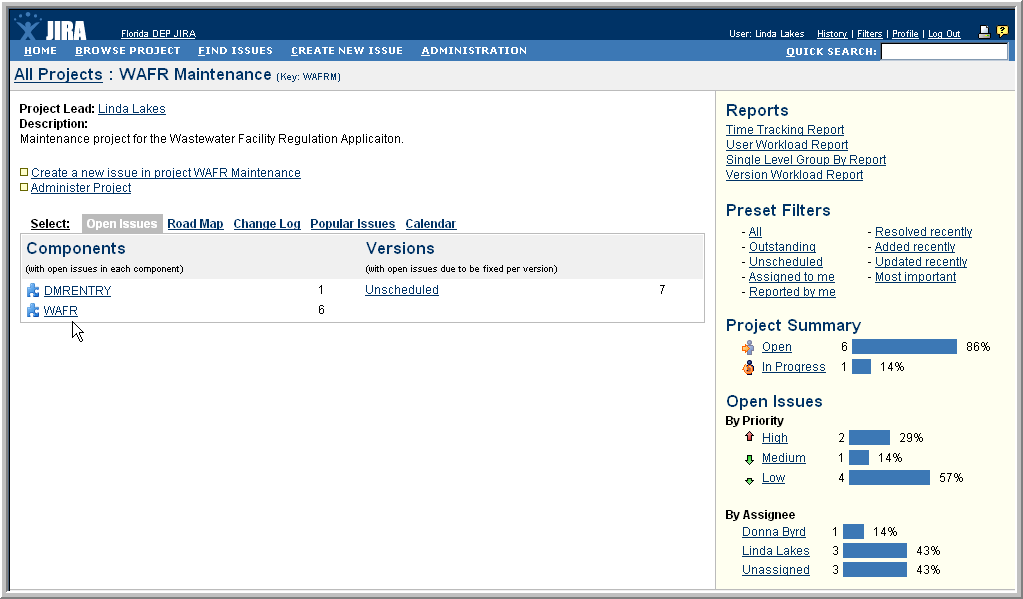
# Updating an Issue

|  |
| --- |
| *Remove this section if users are not allowed to update an issue for your project.* |

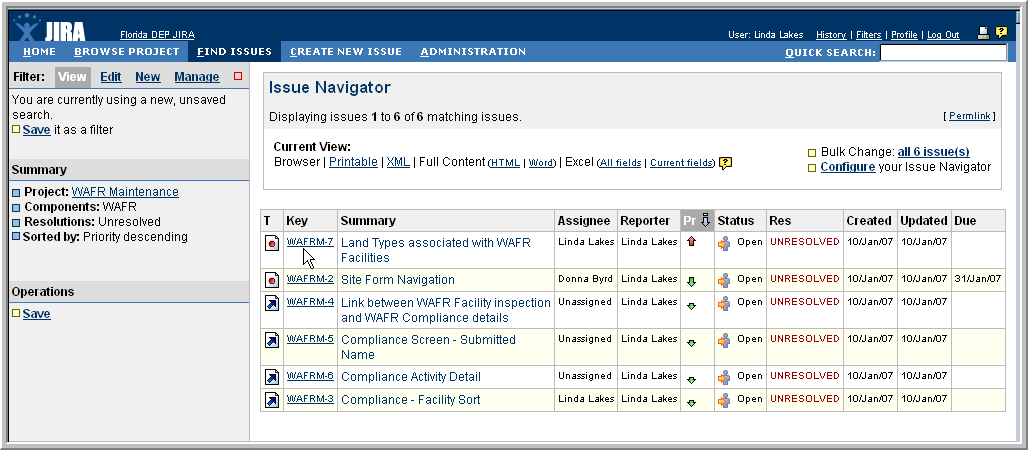
To update an issue, navigate to the Project Summary screen from the Home screen by selecting the project title **[your JIRA project name]**.



Choose the component **[your project component names]** under which the issue is logged by selecting the component name.



Locate the issue that you want to update and select the short name of the issue to access the Issue Details screen.



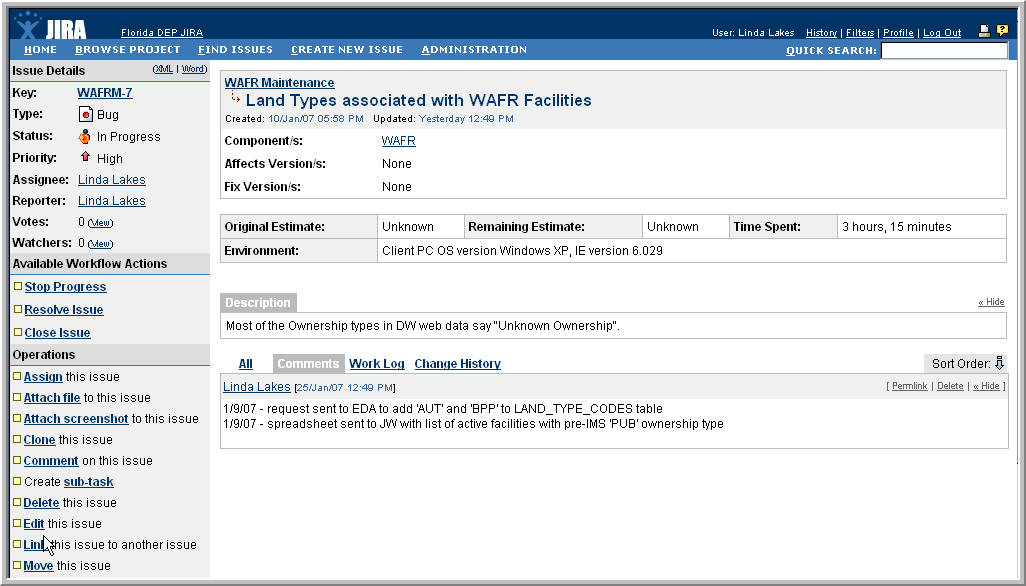
On the left side of the screen is a menu bar with different update options. If no one has begun resolution of the issue, the status will be Open.



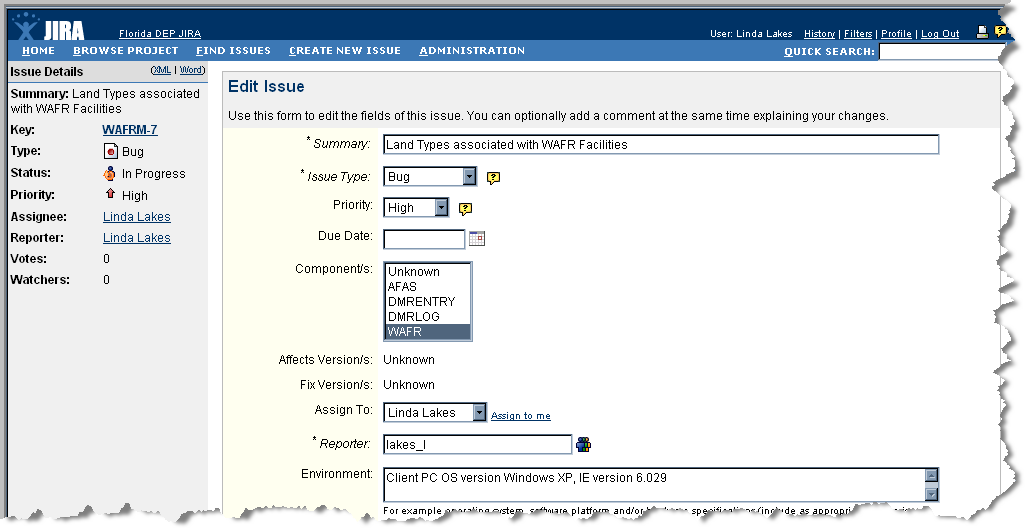
## Editing an Issue

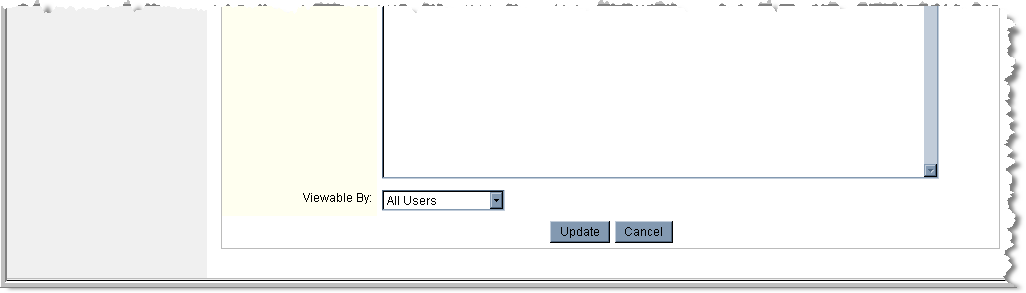
|  |
| --- |
| *Remove this section if users are not allowed to edit an issue for your project.* |

To edit issue information, select the *Edit* menu option under the Operations section.



Enter information related to the issue and select the *Update* button.

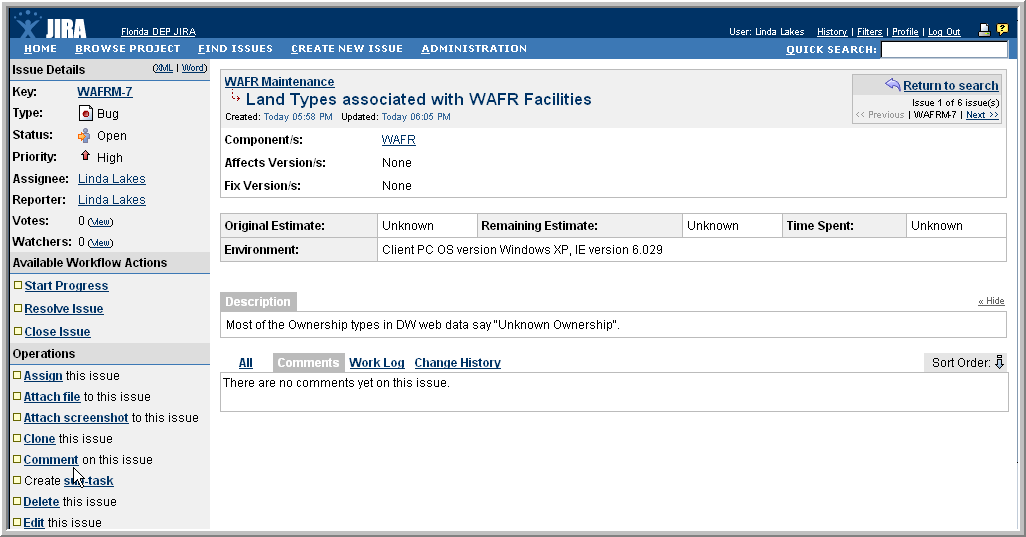




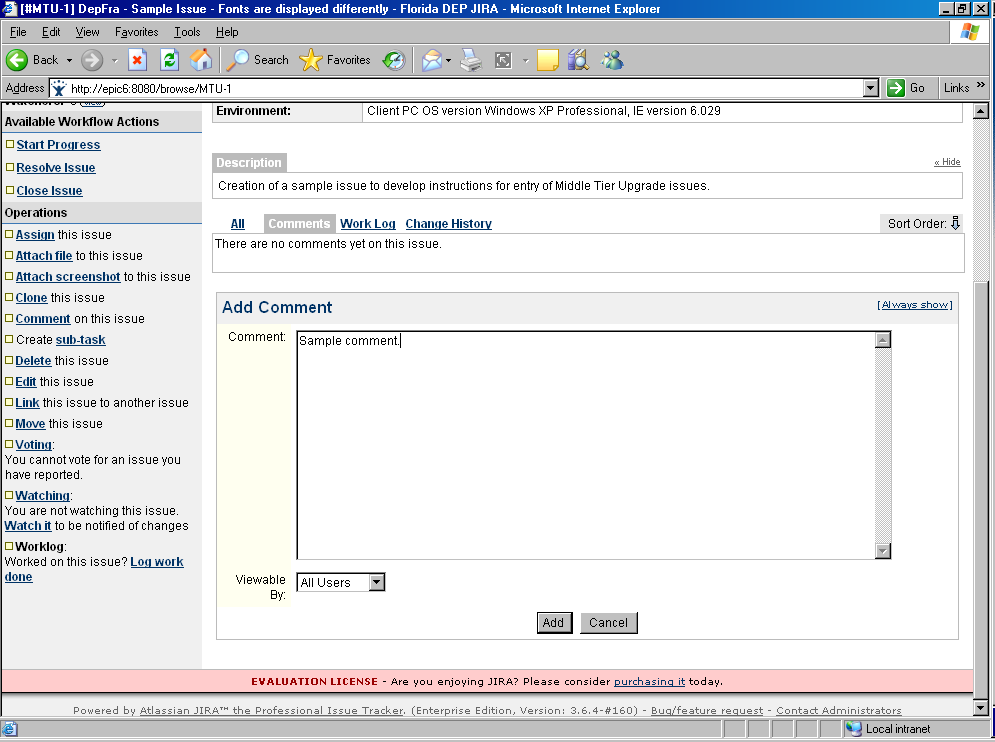
## Adding a Comment to an Issue

|  |
| --- |
| *Remove this section if users are not allowed to add a comment to an issue for your project.* |

To add comments without resolving the issue, select the *Comment* menu option under the Operations section.



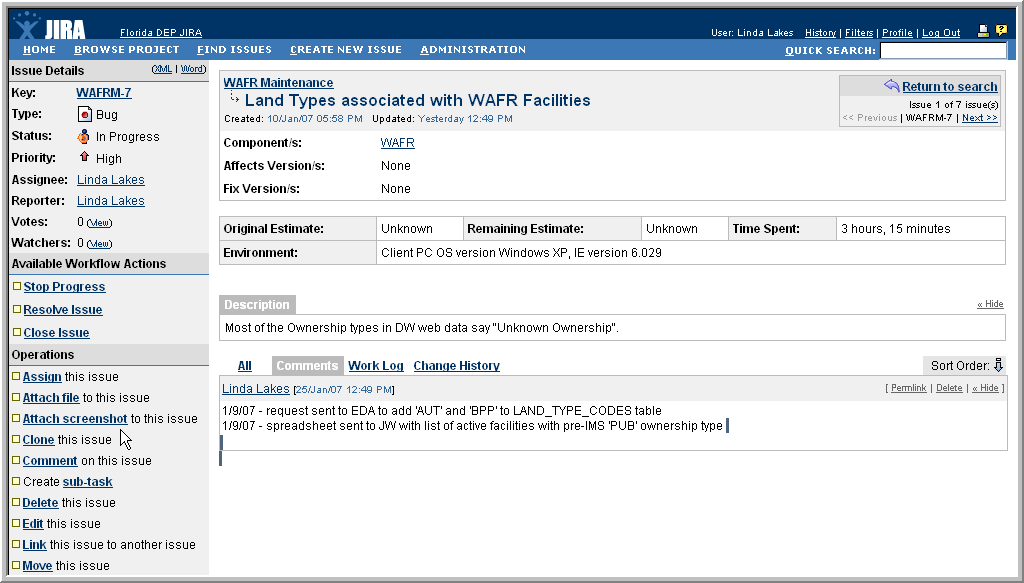
Enter the comment and select the *Add* button.



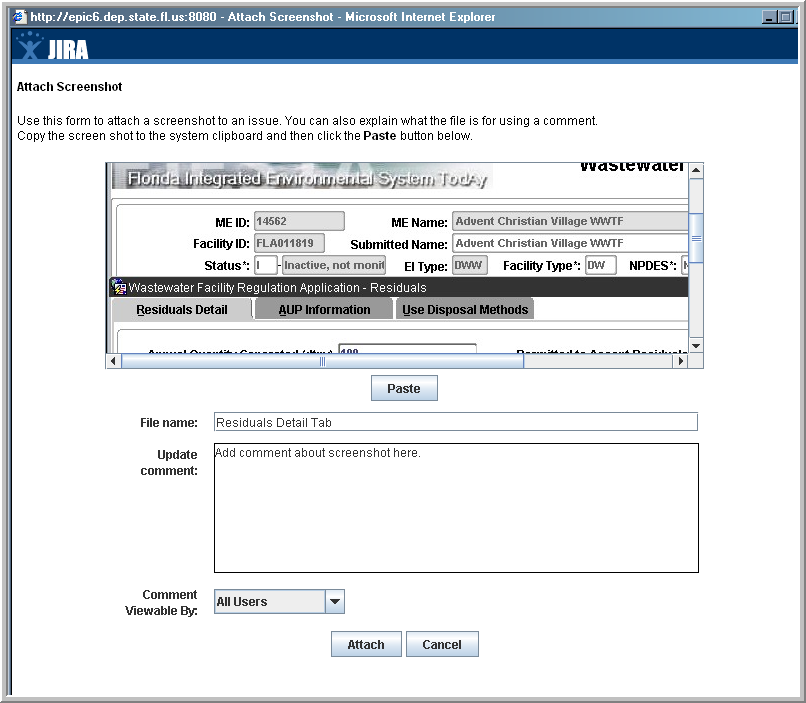
## Attaching a Screenshot to an Issue

|  |
| --- |
| *Remove this section if users are not allowed to add a screenshot to an issue for your project.* |

To attach a screenshot to the issue, select the *Attach screenshot* menu option under the Operations section.



Select the *Paste* button to insert the screenshot on the page. Enter a descriptive file name for the screenshot and add a comment to explain the screenshot. Select the *Attach* button to add the screenshot and associated comment to the issue record.

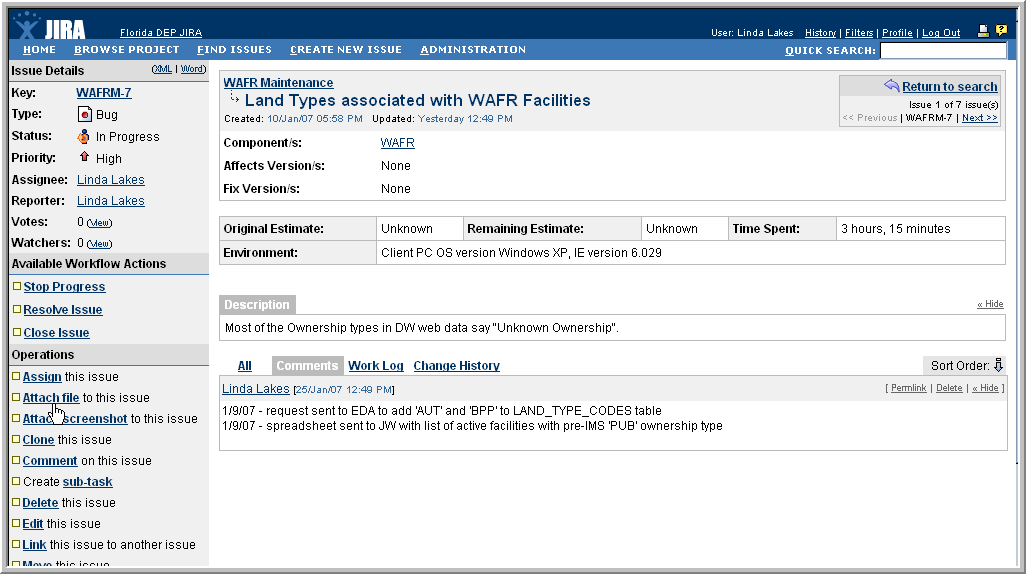


## 

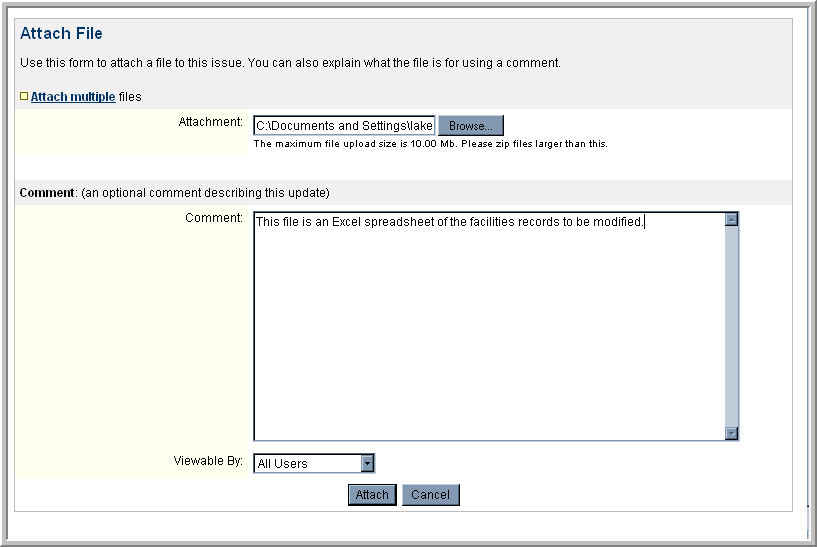
## Attaching a File to an Issue

|  |
| --- |
| *Remove this section if users are not allowed to attach a file to an issue for your project.* |

To attach a file to the issue, select the *Attach file* menu option under the Operations section. If multiple screenshots are needed to illustrate a problem, insert the screenshots into a Word document with comments, and attach the Word file to the issue. The maximum file upload size is 10 Mb. If a file is larger than 10 Mb, zip the file before attaching it to the issue.



Use the *Browse* button to select the file and add a comment to describe the file. Select the *Attach* button to add the file and associated comment to the issue record.



## Life Cycle of an Issue

|  |
| --- |
| *Identify the Status Types used in this project. The JIRA Administrator can assist you with the available options.* |

An issue is created with a status of **Open.** Other statuses for issues include:

* **In Progress** - This issue is being actively worked on by the assignee.
* **Reopened** - This issue was once resolved, but the resolution was deemed incorrect. From here issues are either marked assigned or resolved.
* **Resolved** - A resolution has been taken, and it is awaiting verification by reporter. From here issues are either reopened, or are closed.
* **Closed** - The issue is considered finished, the resolution is correct. Issues which are closed can be reopened.
* **Returned to Development** - Release package has been returned to Development for resolution of issues.

|  |
| --- |
| *Remove this section if you choose not to notify users by a system email for your project. If you choose to notify users, adjust the list below to match the Notification Scheme for your project.* |

For **[your JIRA project name]** project issues, the reporter will be notified by a system email when the status of the project changes. The reporter will also be notified when the issue is:

* Updated
* Assigned
* Resolved
* Closed
* Commented
* Re-opened
* Deleted; or
* Moved

|  |
| --- |
| *Identify the Resolution Types used in this project. The JIRA Administrator can assist you with the available options.* |

An issue may be resolved in many ways, only one of them being **Fixed**. The defined resolutions are listed below.

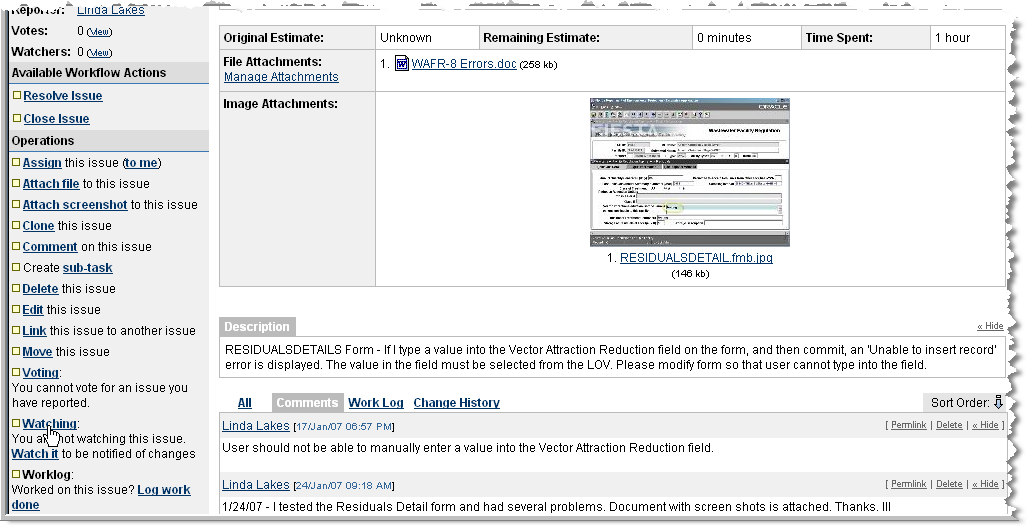
* **Fixed** - A fix for this issue is checked into the tree and tested.
* **Won't Fix** - The problem described is an issue which will never be fixed.
* **Duplicate** - The problem is a duplicate of an existing issue.
* **Incomplete** - The problem is not completely described.
* **Cannot Reproduce** - All attempts at reproducing this issue failed, or not enough information was available to reproduce the issue. Reading the code produces no clues as to why this behavior would occur. If more information appears later, please reopen the issue.
* **Not Determined** - The resolution for this issue has not yet been determined.

When the issue status is **Fixed**, the reported will be asked to re-test the application.

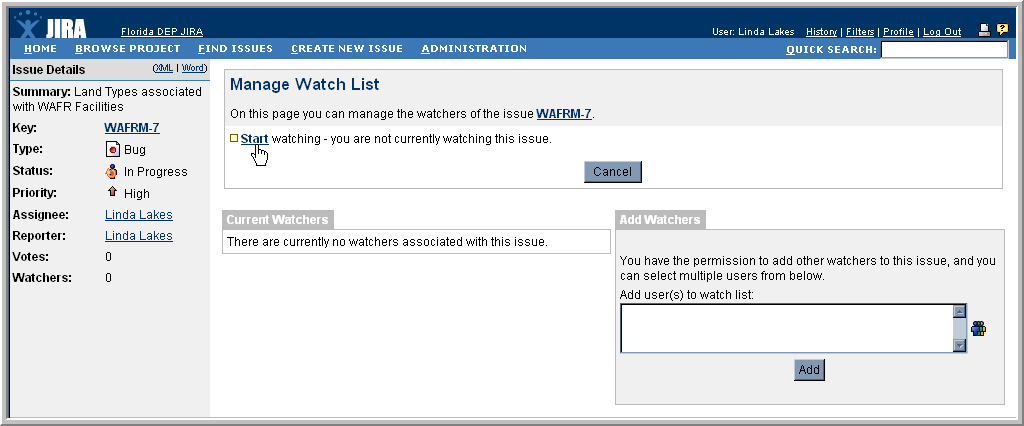
## Watching an Issue

|  |
| --- |
| *Remove this section if you choose not to notify users by a system email for your project.* |

To be notified of changes to an issue, select the *Watching* menu option under the Operations section.



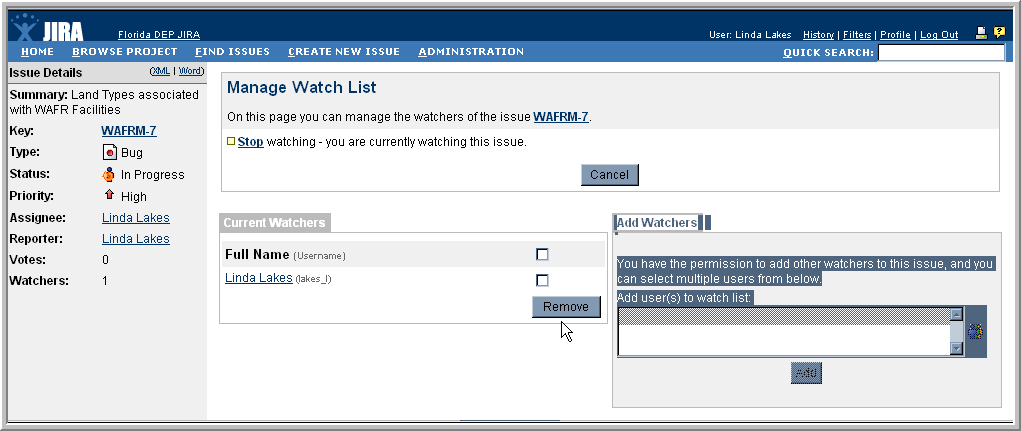
Select the *Start* option.



You will be listed as a current watcher of the issue. Watchers are notified by system email when an issue is:

* Assigned
* Closed
* Commented
* Re-opened
* Deleted; or
* Moved

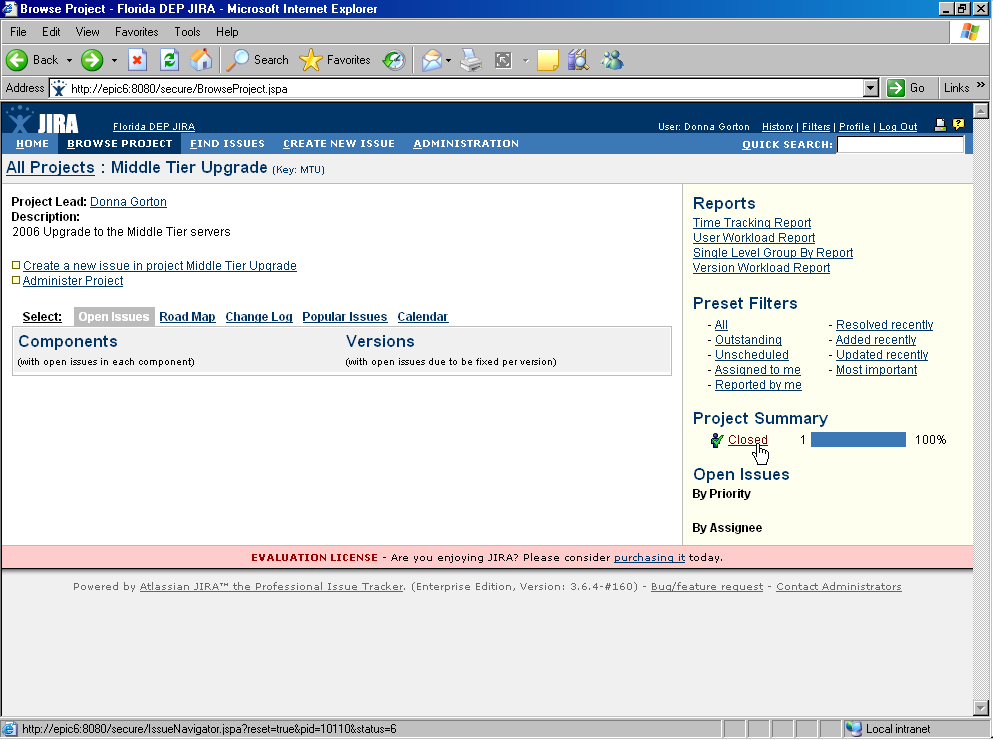
To discontinue watching an issue, select the *Remove* button.



# Closing an Issue

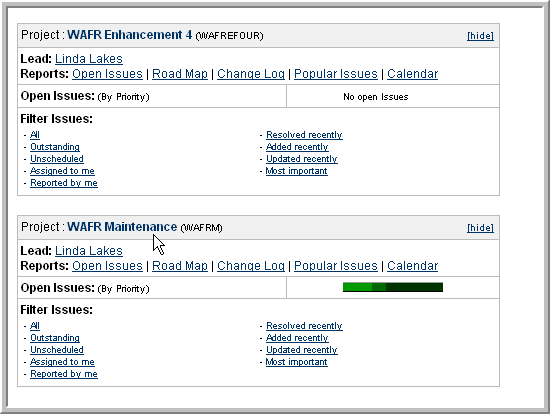
|  |
| --- |
| *Modify this section if you choose not to allow a group of users to close an issue. Indicate the group that is allowed to perform this function.* |

Only the **[your JIRA project name]** Project Lead or delegate may close an issue once it is reported. Closing an issue indicates that the group agrees that the resolution was satisfactory. Issues that are closed may be viewed from the Browse Project screen by selecting *Closed* under the Project Summary section.



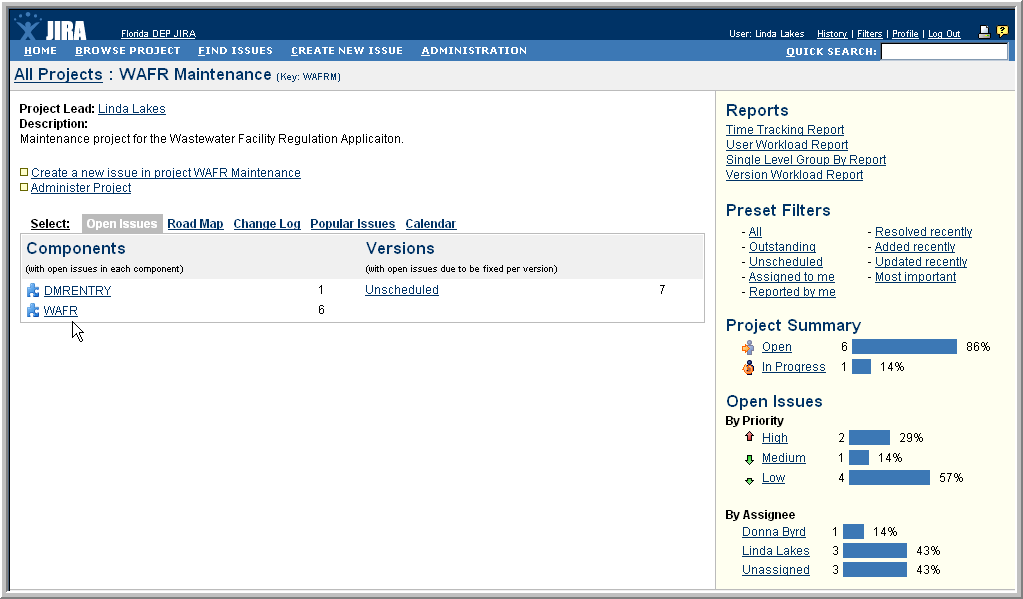
# Searching for an Issue

You may search for an existing issue for a project from the Home screen. The project summary box allows you to access reports and filter issues by clicking hyperlinks. To view all issues, navigate to the Project Summary screen by selecting the project title.

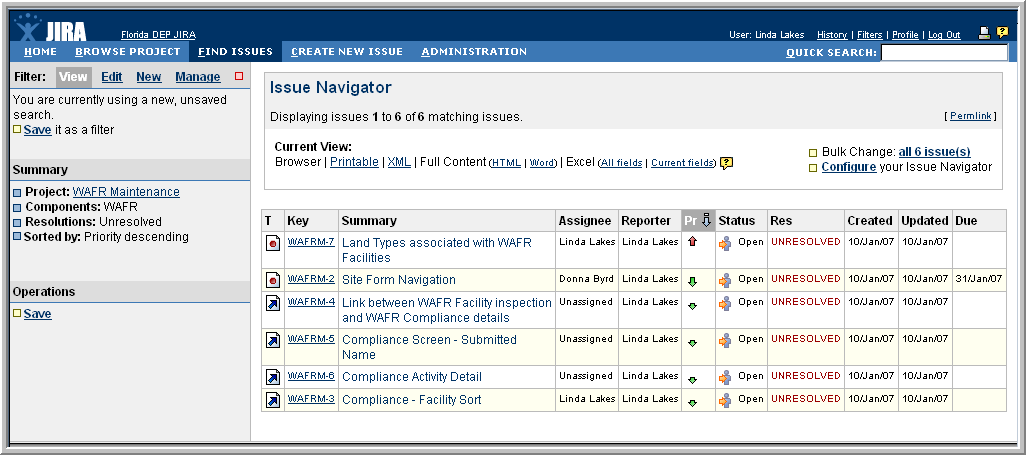


|  |
| --- |
| *Delete this references to “component” if you choose not to add components to your project.* |

Choose the component **[your JIRA project component names]** under which the issue is logged by selecting the component name.



The screen will display all issues logged for this project and component.



# JIRA Support

If you encounter problems while using JIRA for **[your JIRA project name]** project, contact the JIRA administrator using the contact information found on the JIRA logon screen.

1. The name JIRA is derived from GoJIRA, the Japanese name for Godzilla. [↑](#footnote-ref-1)
2. The name JIRA is derived from GoJIRA, the Japanese name for Godzilla. [↑](#footnote-ref-2)