Change History

Version	Date	Prime	Description	
1	12/12/16	Rx196w	Initial Release	
2	12/14/16	Rx196w	Minor editorial corrections	
3	12/19/16	Rx196w	Resolution field is now a required field in all screens; User will	
			need to manually set it to Unresolved upon creation of ticket.	

Create Bug Ticket

To create a Bug ticket, once logged into Jira, click on the Create option at the top of your screen



This will bring up the Create screen, it's pretty intuitive once you see this, but table below will guide you on what fields need to be populated. Fields with a red asterisk (*) are mandatory. If you don't see all the fields, check the Configure fields selected at the top right of the screen and ensure all are selected.

Create issue	[🏶 Configure fi	elds 🔻
Project*	Orange (OR)		<u> </u>
Issue Type*	Bug Some issue types are unavailable due to incompatible field configuration and/or workflor	w associations.	
Summary*			
Description	Style • B I <u>U</u> <u>A</u> • ³ A • Ø • W • ⊨ ⊨ ® •	+ -	~
Component/s*	Start trains to get a list of possible matches or proce down to select	•	
Severity	None Severity of the issue		
Priority	↑ Medium		•
	Create anoth	er Create	Cancel

Bug: Use Case 1: Test originated bug, Code fix provided, Retest successful

The general workflow for Use Case will be:

• Open -> In Progress -> Submitted -> Delivered -> Validation Complete -> Closed

General Flow:

		OPEN		IN PROGRESS	SUBMITTED
General comment: • Upon ticket creation, they will be automatically assigned to Component lead if assignee is not designated		When: Bug issue is created; <i>Open/Unresolved</i> is default state Who: Tester or anyone who wants to report a defect against ECOMP component		When: Bug is moved to <i>In</i> <i>Progress</i> by assignee when work on issue has begun. Resolution field is still <i>Unresolved</i> at this stage Who: Assignee (usually developer)	 When: Bug is moved to Submitted when issue is resolved. If code change, when coding & development testing is completed and code submitted. Resolution should also set to Fixed If not code issue, then Resolution should be set as applicable (e.g., Done if a doc update) Who: Assignee (usually developer)
DELIVERED		VALIDATION COMPLETE		CLOSED	
 When: Bug is moved to <i>Delivered</i> when solution is ready for delivery. If code fix, built is ready for delivery to test team. Notification must be provided of load build availability and ticket assigned back to the originator If not code, follow applicable step and route ticket back to originator Who: Assignee(usually developer or designate depending on team working agreements) updates bug to <i>Delivered</i>, ensures all applicable fields are populated and assigns ticket to Reporter. 	•	 When: Bug is moved to Validation Complete when test completes testing and verifies that bug is fixed. Who: Tester assigned to validate fix moves the bug to Validation Complete and assigns it back to originator for closure if they are not the Reporter. If they are, then can proceed to Closure. 	•	 When: Bug is moved to <i>Closed</i> when originator agrees defect is addressed Who: Originator (Reporter) of ticket. Closed is a final state. Ticket cannot be reopened once closed. 	

Workflow and fields to populate

Jira Status	What Does It Mean?	Who Updates It?	Fields that must be populate
Step 1: Oper	Bug via Create option, populate fie	lds noted below,	, and confirm creation
Open	This is the default status when a Bug is initially created in Jira, the originator is presented with Create Issue screen shown above.	Originator of ticket (Tester)	 The originator of the ticket must provide thorough details regarding the defect. This includes the following (fields with a red asterisk (*) are mandatory): *Project – if you have access to multiple projects *Issue Type (Bug) – make sure you select correct Issue Type *Summary – provide concise/short descriptive summary of the issue Description – detailed description of the issue, with enough information needed by developer to be able to reproduce issue. *Component – identify where the defect is – this will allow proper assignment of the defect Severity and Priority – identify the urgency of the issue – make sure it makes sense, for example a Sev 1 issue with a priority of low or trivial does not make sense. Environment – identify the test environment in which the issue was found

Jira Status	What Does It Mean?	Who Lindates It?	Fields that must be populate
		Opdates it?	• • • •
			 Assignee – if you know who to assign the ticket to, go ahead and enter this person's name. They must be a valid user in the Jira project. If you don't know who to assign it to, leave this field blank; ticket will automatically be assigned to the component lead associated with the identified component on the ticket. Affects Version – identifies the software version in which the bug was found in. Very important to identify this so that developer knows which release the defect is being reported against. Component – identifies where the defect is – this will allow proper assignment of the defect Attachment - Attach applicable logs, screen shot, etc anything that would be helpful to the developer Test Case ID – if the issue is found in the execution of a test case, enter test case ID *Resolution – upon initial creation, this this field to Unresolved
Step 2: Bug is	s now assigned to a developer and w	vork is starting o	n it.
In Progress	The <i>Status</i> field is updated from <i>Open</i> to <i>In Progress</i> when the developer has started work on the issue.	Assignee of Bug issue (Developer)	Developer moves issue from Open to In Progress when they have started working on it by selecting the In Progress option: <pre></pre>
Step 3: Deve	loper has identified the fix for the is	sue and is now r	eady to submit

Jira Status	What Does It Mean?	Who	Fields that must be populate
		Opdates It?	
Submitted	The <i>Status</i> field is updated from <i>In Progress</i> to <i>Submitted</i> when the fix for the defect is submitted into the code repository and ready for a build	Assignee of Bug issue (Developer)	 Developer moves issue from <i>In Progress</i> to <i>Submitted</i> when they have finished their coding/unit testing and are ready to submit to code repository to be picked up in next build. Selecting the Submit option will present the Submit screen to the developer: Edit Assign Comment Closed Submit Ensure following fields are populated: <i>*Resolution</i> – the Resolution field is a required field in the Submit screen. Several options are available, select the one that is applicable to your situation. In the case of bug fixes, the Resolution would be <i>Fixed</i> <i>Fix Version/s</i> – identify the release version of the software in which the fix is provided <i>Build Number</i> – if known at this stage, enter the build number; otherwise, leave blank and enter at next step (when ticket is marked <i>Delivered</i>) <i>Comments</i> - Ensure that appropriate details are entered in the Bug ticket about the nature of the fix. This will help the tester or anyone else looking at the issue to understand the nature of the problem and resolution; identify if additional regression is needed for retest, etc
Step 4: Fixes	have been merged and software bu	ild is ready for d	elivery to test team
Delivered	The Status field is updated from	Assigned of	Developer moves issue from Submitted to
Delivered	Submitted to Delivered when the software build is ready for delivery to the test team	Assignee of Bug issue (Developer)	Developer moves issue from Submitted to Delivered when the software build is available for testing teams to deploy to their lab environment. At this stage, the Jira bug should be properly updated with needed details and Bug Issue assigned back to test team (originator of ticket). Selecting the Deliver option will present the Deliver screen to the developer: Pedit Assign Comment Closed Deliver In-Progress Ensure following fields are populated:

Jira Status	What Does It Mean?	Who Updates It?	Fields that must be populate
			 *Resolution – the Resolution field is a required field in the Submit screen. You have an opportunity to update it again in the Deliver screen if applicable; otherwise, no further action is needed. Fix Version/s – identify the release version of the software in which the fix is provided if not already populated Build Number – enter the build number. If one was entered when ticket was moved to Submit, validate that it's correct. Assignee – assign ticket back to the originator or to the person that must test the fix.
Step 5: Test	team has tested the fix and confirm	that it's working	as expected
Validation Complete	The <i>Status</i> field is updated from <i>Delivered</i> to <i>Validation Complete</i> when testing has successfully completed.	Originator of ticket (Tester)	 Tester moves issue from <i>Delivered</i> to <i>Validation</i> <i>Complete</i> after they successfully validate the bug fix. Selecting the Validation Complete option will present the Validation Complete screen to the tester: <i>Edit</i> Assign <i>Comment Closed</i> Validation Complete Re-Open Ensure following fields are populated/updated as applicable. <i>Comments</i> – Update comments with details of testing and results. <i>Test Case ID</i> – Update test case ID if applicable <i>Assignee</i> – assign ticket back to the originator if different from current assignee
Step 5: All do	one, ticket is closed		
Closed	The <i>Status</i> field is updated from <i>Validation Complete</i> to <i>Closed</i> .	Originator of ticket (Tester)	Tester moves issue from Validation Complete to Closed after they successfully validate the bug fix. Closed is a final state; one ticket is closed, it cannot be moved back to any previous state. If the person who validated the ticket is different than the person who originated the ticket, then the originator should close the ticket after they review the test results.

Bug: Use Case 2: Test originated bug, Code fix provided, Retest Failed

The general workflow for Use Case will be:

• Open -> In Progress -> Submitted -> Delivered -> Re-Open (Status Open, Resolution Fix Failed)

In this use case, the workflow is the same as above, everything proceeds the same up through the Delivered state. At this point, the ticket is Re-opened with Resolution of Fixed Failed and assigned back to the developer and the cycle starts back from Open state.

General Flow:

Jira Status	What Does It Mean?	Who	Fields that must be populate			
		Updates It?				
Stons 1-4.	Same as for Use Case 2					
510051 4.	Steps 1 - 4. Same as for Use Case 2					
Step 5: Test	team has tested the fix and the fix F	ailed				
Open	The <i>Status</i> field is updated from <i>Delivered</i> back to Open and Resolution is marked as Fix Failed.	Originator of ticket (Tester)	 Tester moves issue from <i>Delivered</i> back to Open by selecting the Re-Open option: <i>i</i> Edit Assign <i>Comment</i> Closed Validation Complete Re-Open Ensure following fields are populated/updated as applicable. <i>Comments</i> – Update comments with details of testing and results. <i>Attachment</i> - Attach applicable logs, screen shot, etc anything that would be helpful to the developer <i>Test Case ID</i> – Update test case ID if applicable <i>Assignee</i> – assign ticket back to the Developer. If you're not sure who that is, look in the History of the ticket and see who move ticket to Submitted state. For example: Activity <i>Comments</i> Vork log History Activity <i>Comments</i> Vork log History Activity <i>Status</i> Open[1] In Progress[3] Submitted [1010] 			
Jila issue goe	es through same now starting with s	neh Tagam.				