FAQ

- Testing Process
  - Is it possible to use Time Tracking with Xray? Is it possible to log time associated with the running of Test Runs?
- Upgrading
  - What should I do to update/upgrade my current version of Xray?
- Requirements
  - I cannot find the "Requirement" issue type. Why?
  - I'm unable to see the "Test Coverage" panel and the "Requirement Status" in my requirements. Why?
  - The "Requirement Status" custom field does not show the "expected" value. Why?
  - The status of my requirements always shows as Not Run, independently of the results I have recorded
- Tests
  - Can I use a Test for testing different system versions?
  - How do I associate a Test with a project version?
  - Do you support versioning of Tests?
  - How do I define which Tests should be used for regression testing?
  - Can I link Tests to other Tests?
  - The "TestRunStatus" custom field does not show the "expected" value. Why?
- Test Sets
  - Can I clone a Test Set? Will that clone the Tests that belong to the Test Set?
- Test Executions
  - How do I create a Test Execution for all failed tests (or other test status) of a previous execution?
  - Can I run a Test Execution multiple times?
  - Do I have to use Test Environments?
  - Do I have to execute tests in all Test Environments?
  - Can I take screenshots and attach them when executing tests?
- Test Plans
  - A “Test Plan” seems like a group of Test Executions or a group of Tests.
  - Can I clone “Test Plans”?
  - Can I have multiple Test Plans for a version?
  - Do I have to use "Test Plans"?
- Agile Integration
  - Can I add a Sub-Test Execution to multiple Requirement issues?
  - Do I have to associate all Tests that are testing a Requirement with a Sub-Test Execution?
- Configuration
  - Is it possible to use custom workflows for Test-related issues? Does Xray install or enforce any workflow?
- Licenses
  - Where can I find the pricing for Xray and licensing information?
  - How do I install a license provided by Xpand IT?
  - How can I replace my Marketplace license with one provided by Xpand IT?
  - What happens when my paid license expires?
  - Do I need a license?
  - Can I renew my trial license?
- Installation
  - What Jira versions do you support?
  - Do you support Jira Data Center (i.e., high availability)?
  - How do I roll back to a previous version of Xray?
  - What happens if I uninstall Xray?
  - How do I enable debug logging?
- Import/Export
  - Import JSON execution results from Cucumber returning error
  - Is it possible to import Tests and link them to other issues (e.g., requirements)?
  - Can I migrate my Tests from one Jira server to another?
  - How can I migrate my legacy executions to Jira?
- Integrations
  - Can I trigger/start Jenkins/Bamboo builds from Xray?
- JQL functions
  - When searching for issues, custom fields (e.g., TestRunStatus) do not return the correct values.
- Jira startup
  - Xray is disabled upon Jira startup
- Project archiving
  - Can I know the progress of archiving or restoring Xray entities when archiving or restoring the Project?
- Contact
  - Can I send you guys an email?
  - I would like a demo. Is that possible?

Testing Process

Is it possible to use Time Tracking with Xray? Is it possible to log time associated with the running of Test Runs?

Yes. Please refer to the Time Tracking documentation.

Upgrading

What should I do to update/upgrade my current version of Xray?

First, please read all the release notes in Release Notes for all versions, including bug fixes, between your current and your target version; most times updating is as simple as clicking the "Update" button next to your app, in the dedicated apps section within your Jira administration (please see detailed instructions Installation page).

Sometimes, if highlighted in the release notes, you may need to perform the re-calculation of custom fields and/or re-index.

If you are using Jira Data Center, you must restart all nodes after installing the new version of Xray.

Requirements

I cannot find the "Requirement" issue type. Why?

There is no "Requirement" issue type installed by Xray. In fact, Xray does not install any requirement-related issue types. If you had the opportunity to attend one of our webinars, particularly the Xray walkthrough, you will learn that with Xray, you are able to define which issue types Xray should consider as requirements so you can cover them with Tests. Besides the standard "Story" and "Epic" Jira issue types, Jira allows you to create your own issue types. You just need to configure Xray properly, as detailed here, to let Xray internally treat those issue types as being requirements.

I'm unable to see the "Test Coverage" panel and the "Requirement Status" in my requirements. Why?

You need to ensure that:

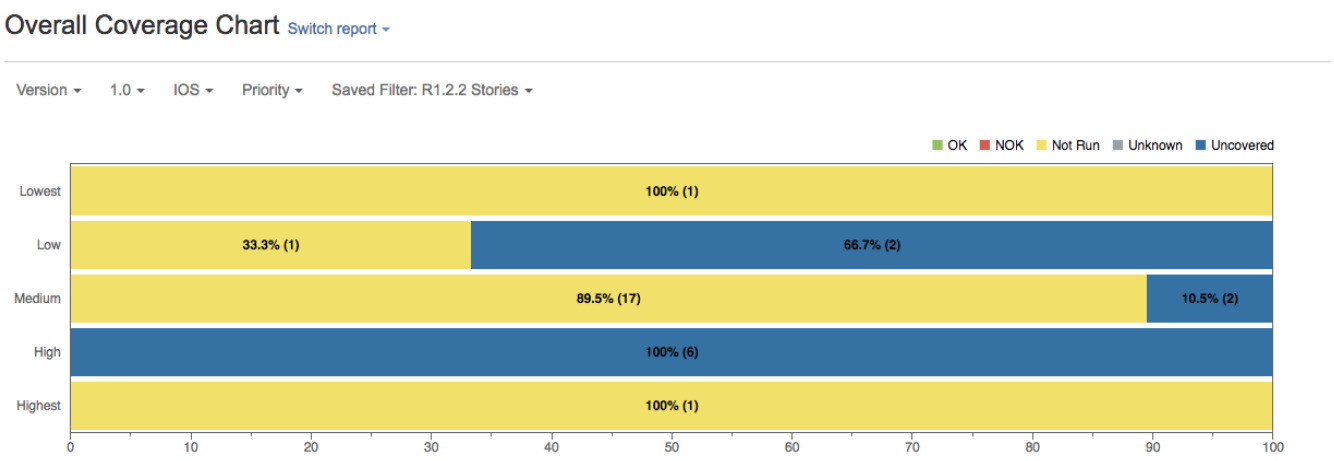
1. your project is defined as a requirements project, and
2. your issue type is configured to be a requirement issue type.

Both can be achieved in Xray's administration (see Quick Setup for instructions), although the first one may also be done in the project settings page.

The "Requirement Status" custom field does not show the "expected" value. Why?

The "Requirement Status" is a **calculated** field, so it shows the information for some version depending on a Xray setting available in Custom Field Preferences. Currently, it is not possible to specify explicitly the version that you want to calculate the coverage on. Please read Using custom fields for a deeper explanation.

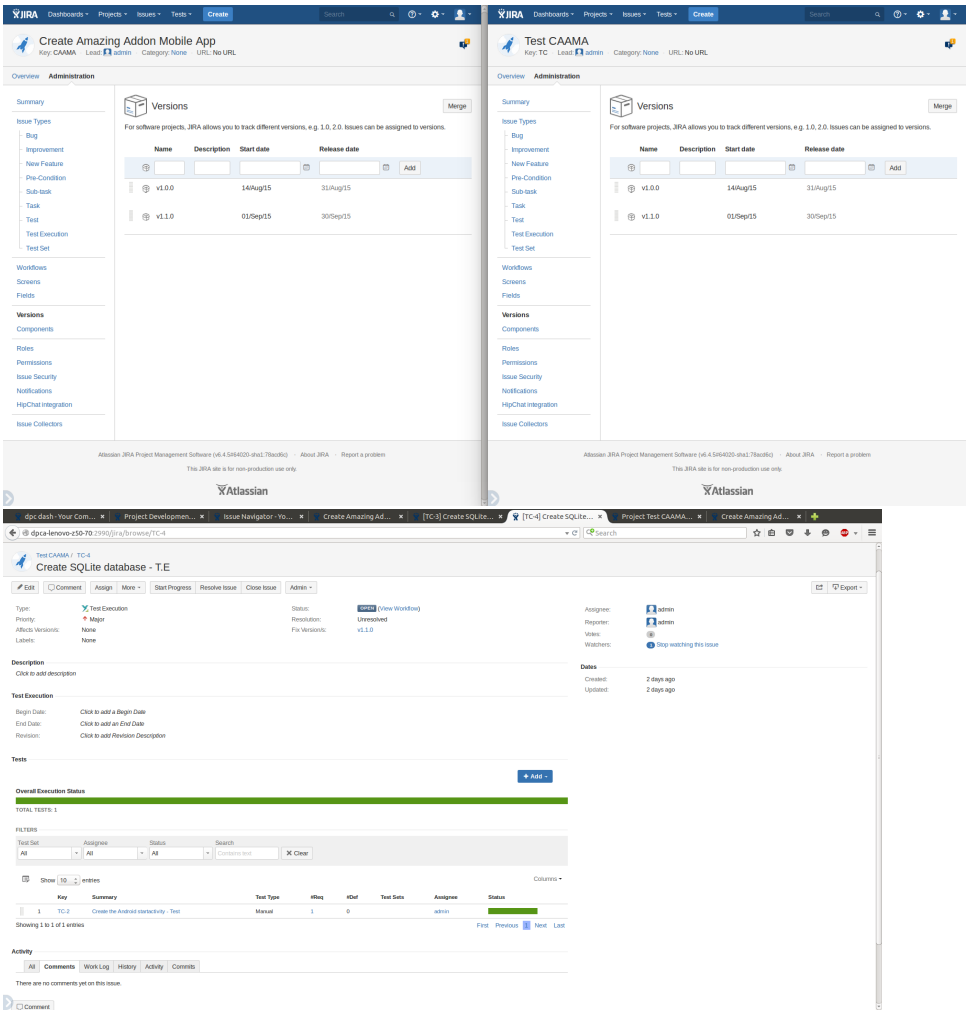
The status of my requirements always shows as Not Run, independently of the results I have recorded



You might be using environments on Test Executions that are affecting the aggregated status for the Tests. If you have a couple of Test Executions, one with a specific environment (e.g., Android) and another without any environment, then Xray will also consider the empty environment when calculating the aggregated status (this is, of course, if all Test Executions are within the same Version or Test Plan).

You have not configured your versions accordingly. Remember that requirements coverage is calculated with Test Execution and/or Test Set for a Requirement Fix version.

You must have the same version name between projects. The Test Execution **fix version** must have the same value as the Requirement Issue **fix version.**



Tests

Can I use a Test for testing different system versions?

Yes. A Test is like a test template. You can use it in multiple Test Sets or Test Executions, which may be assigned with different versions.

How do I associate a Test with a project version?

The association of a Test with a project version is not direct. Your Tests are written in order to validate some requirement, that itself belongs to a specific version. Although a Test may be associated with a requirement of a specific version, you are able to run Tests for the versions you like. In terms of requirement coverage, you have different behaviors to choose from in the administration settings (see Requirement Coverage Strategy). You may also to have a look at Tips for implementing Test Versioning.

Do you support versioning of Tests?

Please refer to Tips for implementing Test Versioning.

How do I define which Tests should be used for regression testing?

Please take a look at Tips for organizing tests.

Can I link Tests to other Tests?

Yes. You can create normal links between Test issues, as you would do for any standard Jira issue. Xray will not process those relationships in any special way; it will ignore them unless they are of "tests" or "tested by" relationships.

The "TestRunStatus" custom field does not show the "expected" value. Why?

The "TestRunStatus" is a **calculated** field, so it shows the information for some version, depending on a Xray setting available in Custom Field Preferences. Currently, it is not possible to specify explicitly the version that you want to calculate the coverage on. Please read Using custom fields for a deeper explanation.

Test Sets

Can I clone a Test Set? Will that clone the Tests that belong to the Test Set?

You can clone a Test Set and the cloned Test Set will have the same Test Sets that were in the original Test Set. The tests themselves are not cloned. If you really want to clone a bulk list of tests, then you must clone them manually (you may find add-ons for this purpose), and then associate them to the new Test Set. In fact, whenever you clone Test Sets, Test Executions or Test Plans, the Tests that they contain are not cloned; just the internal association.

Test Executions

How do I create a Test Execution for all failed tests (or other test status) of a previous execution?

You can do it easily within the Test Plan screen via the "Create Test Execution" button. You may also create a new Test Execution for some of the tests listed in the table below the Filters, by pressing the bulk icon blocked URL and then pick the relevant tests (you may use a filter to help you out), and then choose the Create in the actions icon blocked URL, available on the right side of the table.

Can I run a Test Execution multiple times?

Kind of. A specific Test Execution represents an execution task of a list of tests, for a specific version/revision, done at a specific time. With Xray, the normal way of executing a set of tests multiple times is by creating a Test Execution each time you want to execute the tests. All you need to do is clone an existing Test Execution; the newly-created Test Execution will contain the same list of tests, in clean state since you haven't executed them yet.

Do I have to use Test Environments?

No, but if you execute the same tests in different environments, you certainly will find it as the most useful and accurate way of testing a requirement.

Do I have to execute tests in all Test Environments?

No. It's up to you to organize your Tests per environment. This is the reason we use the Test Environment Custom Field as a label.

Can I take screenshots and attach them when executing tests?

Xray does not provide this feature out-of-the-box by itself. However, you may use a simple tool dedicated to taking screenshots (e.g., LightShot) and easily attach the screenshot to the evidences during the execution of the test.

Test Plans

A "Test Plan" seems like a group of Test Executions or a group of Tests.

A bit of both, in fact. A Test Plan contains a list of Tests. For calculating the status of each test, it takes into account test runs from Test Executions explicitly linked to the Test Plan, through the field "Test Plan".

Can I clone "Test Plans"?

Yes. The list of related Tests is copied; the Tests are not cloned.

Can I have multiple Test Plans for a version?

Yes, if you wish to track different groups of Tests (e.g., regression tests *versus* tests for new requirements).

Do I have to use "Test Plans"?

You don't have to, but you will find great advantages if you start using them.

Agile Integration

Can I add a Sub-Test Execution to multiple Requirement issues?

A sub-task can only have one parent issue, so it won't be possible to have multiple requirement issues with the same Sub-Test Execution. However, you can add Tests from another Requirement issue to a Sub-Test Execution. You'll be able to execute them, but you are going to have the same behavior as the Test Execution, while using Agile Boards. So, if you intend to track the executions in the Agile Board, we recommend that you have a Sub-Test Execution for each Requirement issue.

Do I have to associate all Tests that are testing a Requirement with a Sub-Test Execution?

The Tests already linked with the Requirement issue will be automatically added to the Sub-Test Execution. You can remove them if you want. If you add a new Test to the Requirement issue after creating the Sub-Test Execution, then you'll need to add them manually.

Configuration

Is it possible to use custom workflows for Test-related issues? Does Xray install or enforce any workflow?

Xray does not install any "default" workflow for the issues it provides. You may define whatever worfklow you think fits best your scenario/organization. Xray is workflow-aware though, and provides some configurations specifically for that purpose (see Global Preferences page).

Licenses

Where can I find the pricing for Xray and licensing information?

More information can be found in the Xray section of the Atlassian's Marketplace.

How do I install a license provided by Xpand IT?

Once you have a license provided by Xpand IT, perform the following steps:

- In your Jira Administration, select Manage Add-Ons.
- Go to Xray Configuration > License Management.
- Select "Click here to install a license directly from Xpand IT".
- Enter your license in the text box. Remember, this shoud be a license provided by Xpand IT, not Atlassian Marketplace.
- Click Add.
- Your Xpand IT license should be installed.

How can I replace my Marketplace license with one provided by Xpand IT?

If you need to replace your Marketplace license with a license provided by Xpand IT, perform the following steps:

- In your Jira Administration, select Manage Add-Ons.
- Click on "Xray". Detail about the app will be shown.
- Edit the license key, and remove all the text, then click Update. Your marketplace license has been removed.
- Go to Xray Configuration > License Management.
- Select "Click here to install a license directly from Xpand IT".
- Enter your license in the text box. Remember, this shoud be a license provided by Xpand IT, not Atlassian Marketplace
- Click Add.
- Your Xpand IT license should be installed.

What happens when my paid license expires?

If you have a paid license and it expires, you won't be able to upgrade Xray. You won't also be able to access our support, but you still be able to fully use Xray.

Do I need a license?

Yes. Even for trying Xray, you need a trial license. More information can be found in the Xray section in Atlassian's Marketplace.

Can I renew my trial license?

Yes, up to 5 times. More information can be found in the Xray section in Atlassian's Marketplace.

Installation

What Jira versions do you support?

Please see our Installation page.

Do you support Jira Data Center (i.e., high availability)?

Yes.

How do I roll back to a previous version of Xray?

If you need to roll back your installation to a previous version, perform the following steps:

- Uninstall your current version of the app. Navigate to your Jira administration > Manage Add-ons > Xray.
- Click Uninstall.
- Download the version that you want from https://marketplace.atlassian.com/plugins/com.xpandit.plugins.xray/versions (click on the version and choose Download this version).
- In the Manage Add-Ons page, select Upload Add-on and browse the file downloaded in the previous step.
- The previous version of the app should now be installed.

What happens if I uninstall Xray?

If for some reason you need to uninstall Xray, you can do it safely. No relevant data is removed from Jira. This means that Xray issue types, custom fields, Xray settings and all recorded information related with testing is kept.

Note: If you uninstall and install Xray once again, nothing special should happen. In other words, it should be idempotent.

How do I enable debug logging?

If you need to enable logging for the app, perform the following steps:

- Go to the Jira Administration > System.
- Select Logging & Profiling.
- In Default Loggers, click "Configure logging level for another package".
- Set the Package name to **com.xpandit** and Logging Level to **DEBUG.**
- Click Add.

Once you performed the previous steps, you should recreate your problem and retrieve the **atlassian-jira.log** file.

Import/Export

Import JSON execution results from Cucumber returning error

If you get a message like : “Error importing execution results to database: User XXX does not have permission to execute test execution with key TEST-XX”

- Ensure that the User who uploaded the result file has the RESOLUTION permission

- Ensure that Test Execution is not in a Workflow status marked as non-executable in Xray "Global Preferences"

Is it possible to import Tests and link them to other issues (e.g., requirements)?

Yes. You can add one or more columns in your CSV and then map them to "Link Tests"'. Please see the documentation in Importing Manual Tests using Test Case Importer.

Can I migrate my Tests from one Jira server to another?

Yes, with some constraints. Please see the proper section within the documentation in Importing Tests with (CSV).

How can I migrate my legacy executions to Jira?

The built-in Test Case Importer is only able to import Test specifications and not results. There is no immediate way to achive what you need. However, there are some options.

One way is to use the REST API; this will require some development effort on your side and will also require that you have the Tests already created in Jira.
Another option is to use the built-in Jira CSV importer (not Xray's Test Case Importer) to create the Test Executions with the Tests. However, you would still need to use the REST API in order to submit the results for the Tests contained in those Test Executions.

Please see the proper section within the documentation in Importing Tests with (CSV).

Integrations

Can I trigger/start Jenkins/Bamboo builds from Xray?

Currently, this is not possible.

JQL functions

When searching for issues, custom fields (e.g., TestRunStatus) do not return the correct values.

Columns presented in the results do not take into account the context of the JQL query. The TestRunStatus custom field will always present the same result independent of the JQL query.

Let's say you want to see the tests of a given Test Execution, using *testExecutionTests(...)*, and their results. If **Test A** is executed by **TE1** and **TE2** and you're filtering by **TE2,** then you will get the "global" status calculated, and this can be the value in **TE1** if **TE1** is the latest test run.

Jira startup

Xray is disabled upon Jira startup

If you get a log message like:

com.atlassian.cache.CacheException: java.lang.IllegalArgumentException: Unknown icon type 'issuetype'

```
    2017-06-27 06:18:42,964 JIRA-Bootstrap ERROR      [c.x.raven.issuetype.RavenIssueTypeManagerImpl] Error ocurred installing Xray issue types!

com.atlassian.cache.CacheException: java.lang.IllegalArgumentException: Unknown icon type 'issuetype'

                at com.atlassian.cache.memory.DelegatingCachedReference.get(DelegatingCachedReference.java:83)

                at com.atlassian.jira.config.DefaultConstantsManager.getAllIssueTypeObjects
```

```
(DefaultConstantsManager.java:636)

                at com.xpandit.raven.service.impl.RavenIssueTypeServiceImpl.findIssueType(Unknown Source)

                at com.xpandit.raven.issuetype.RavenIssueTypeManagerImpl.a(Unknown Source)

                at com.xpandit.raven.issuetype.RavenIssueTypeManagerImpl.a(Unknown Source)

                at com.xpandit.raven.RavenSetupHandler.setup(Unknown Source)

                at com.xpandit.raven.RavenSetupHandler.onPluginEnabled(Unknown Source)

                ... 2 filtered

                at java.lang.reflect.Method.invoke(Method.java:498)

                at com.atlassian.event.internal.SingleParameterMethodListenerInvoker.invoke
(SingleParameterMethodListenerInvoker.java:36)

                at com.atlassian.event.internal.AsynchronousAbleEventDispatcher$1$1.run
(AsynchronousAbleEventDispatcher.java:48)

                at com.google.common.util.concurrent.MoreExecutors$DirectExecutorService.execute(MoreExecutors.
java:299)

                at com.atlassian.event.internal.AsynchronousAbleEventDispatcher.dispatch
(AsynchronousAbleEventDispatcher.java:107)

                at com.atlassian.event.internal.EventPublisherImpl.invokeListeners(EventPublisherImpl.java:160)

                at com.atlassian.event.internal.EventPublisherImpl.publish(EventPublisherImpl.java:79)

                at com.atlassian.plugin.event.impl.DefaultPluginEventManager.broadcast
(DefaultPluginEventManager.java:73)

                at com.atlassian.plugin.manager.DefaultPluginManager.broadcastIgnoreError(DefaultPluginManager.
java:2122)

                at com.atlassian.plugin.manager.DefaultPluginManager.enableDependentPlugins
(DefaultPluginManager.java:1261)

                at com.atlassian.plugin.manager.DefaultPluginManager.addPlugins(DefaultPluginManager.java:1215)

                at com.atlassian.jira.plugin.JiraPluginManager.addPlugins(JiraPluginManager.java:152)

                at com.atlassian.plugin.manager.DefaultPluginManager.earlyStartup(DefaultPluginManager.java:
597)

                at com.atlassian.jira.plugin.JiraPluginManager.earlyStartup(JiraPluginManager.java:120)

                at com.atlassian.jira.ComponentManager$PluginSystem.earlyStartup(ComponentManager.java:641)

                at com.atlassian.jira.ComponentManager.quickStart(ComponentManager.java:195)

                at com.atlassian.jira.ComponentManager.start(ComponentManager.java:164)

                at com.atlassian.jira.upgrade.PluginSystemLauncher.start(PluginSystemLauncher.java:43)

                at com.atlassian.jira.startup.DefaultJiraLauncher.lambda$postDbLaunch$2(DefaultJiraLauncher.
java:150)

                at com.atlassian.jira.config.database.DatabaseConfigurationManagerImpl.doNowOrEnqueue
(DatabaseConfigurationManagerImpl.java:298)

                at com.atlassian.jira.config.database.DatabaseConfigurationManagerImpl.
doNowOrWhenDatabaseActivated(DatabaseConfigurationManagerImpl.java:194)

                at com.atlassian.jira.startup.DefaultJiraLauncher.postDbLaunch(DefaultJiraLauncher.java:141)

                at com.atlassian.jira.startup.DefaultJiraLauncher.lambda$start$0(DefaultJiraLauncher.java:103)
```

```
                at com.atlassian.jira.util.devspeed.JiraDevSpeedTimer.run(JiraDevSpeedTimer.java:31)

                at com.atlassian.jira.startup.DefaultJiraLauncher.start(DefaultJiraLauncher.java:101)

                at com.atlassian.jira.startup.LauncherContextListener.initSlowStuff(LauncherContextListener.
java:149)

                at java.lang.Thread.run(Thread.java:745)

Caused by: java.lang.IllegalArgumentException: Unknown icon type 'issuetype'

                at com.atlassian.jira.plugin.icon.IconTypeDefinitionFactoryImpl.getDefaultSystemIconFilename
(IconTypeDefinitionFactoryImpl.java:45)

                at com.atlassian.jira.avatar.AvatarManagerImpl.loadDefaultAvatarId(AvatarManagerImpl.java:488)

                at com.atlassian.jira.avatar.AvatarManagerImpl.lambda$getDefaultAvatarId$3(AvatarManagerImpl.
java:509)

                at com.atlassian.vcache.internal.core.DefaultRequestCache.lambda$get$15(DefaultRequestCache.
java:43)

                at java.util.HashMap.computeIfAbsent(HashMap.java:1126)

                at com.atlassian.jira.avatar.AvatarManagerImpl.getDefaultAvatarId(AvatarManagerImpl.java:509)

                at com.atlassian.jira.avatar.AvatarManagerImpl.getDefaultAvatarId(AvatarManagerImpl.java:482)

                at com.atlassian.jira.issue.issuetype.IssueTypeImpl.getAvatarOrDefault(IssueTypeImpl.java:39)

                at com.atlassian.jira.issue.issuetype.IssueTypeImpl.getAvatar(IssueTypeImpl.java:32)

                at com.atlassian.jira.config.DefaultIssueConstantFactory.createIssueType
(DefaultIssueConstantFactory.java:64)

                at com.atlassian.jira.config.DefaultIssueConstantFactory.createIssueType
(DefaultIssueConstantFactory.java:84)

                at java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:193)

                at java.util.ArrayList$ArrayListSpliterator.forEachRemaining(ArrayList.java:1374)

                at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:481)

                at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:471)

                at java.util.stream.ForEachOps$ForEachOp.evaluateSequential(ForEachOps.java:151)

                at java.util.stream.ForEachOps$ForEachOp$OfRef.evaluateSequential(ForEachOps.java:174)

                at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)

                at java.util.stream.ReferencePipeline.forEach(ReferencePipeline.java:418)

                at com.atlassian.jira.config.DefaultConstantsManager.loadIssueTypeCache(DefaultConstantsManager.
java:799)

                at com.atlassian.cache.memory.MemoryCacheManager$1$1.load(MemoryCacheManager.java:129)

                at com.atlassian.cache.memory.MemoryCacheManager$1$1.load(MemoryCacheManager.java:105)

                at com.google.common.cache.LocalCache$LoadingValueReference.loadFuture(LocalCache.java:3527)

                at com.google.common.cache.LocalCache$Segment.loadSync(LocalCache.java:2319)

                at com.google.common.cache.LocalCache$Segment.lockedGetOrLoad(LocalCache.java:2282)

                at com.google.common.cache.LocalCache$Segment.get(LocalCache.java:2197)

                at com.google.common.cache.LocalCache.get(LocalCache.java:3937)
```

```
                at com.google.common.cache.LocalCache.getOrLoad(LocalCache.java:3941)

                at com.google.common.cache.LocalCache$LocalLoadingCache.get(LocalCache.java:4824)

                at com.google.common.cache.LocalCache$LocalLoadingCache.getUnchecked(LocalCache.java:4830)

                at com.atlassian.cache.memory.DelegatingCachedReference.getUnderLock(DelegatingCachedReference.
java:93)

                at com.atlassian.cache.memory.DelegatingCachedReference.get(DelegatingCachedReference.java:78)

                ... 35 more

2017-06-27 06:18:42,980 JIRA-Bootstrap ERROR      [c.xpandit.raven.RavenSetupHandler] Xray add-on
initialization failed! Disabling Xray add-on...

com.atlassian.jira.exception.CreateException: com.atlassian.cache.CacheException: java.lang.
IllegalArgumentException: Unknown icon type 'issuetype'

                at com.xpandit.raven.issuetype.RavenIssueTypeManagerImpl.a(Unknown Source)

                at com.xpandit.raven.RavenSetupHandler.setup(Unknown Source)

                at com.xpandit.raven.RavenSetupHandler.onPluginEnabled(Unknown Source)

                at sun.reflect.GeneratedMethodAccessor136.invoke(Unknown Source)

                at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

                at java.lang.reflect.Method.invoke(Method.java:498)

                at com.atlassian.event.internal.SingleParameterMethodListenerInvoker.invoke
(SingleParameterMethodListenerInvoker.java:36)

                at com.atlassian.event.internal.AsynchronousAbleEventDispatcher$1$1.run
(AsynchronousAbleEventDispatcher.java:48)

                at com.google.common.util.concurrent.MoreExecutors$DirectExecutorService.execute(MoreExecutors.
java:299)

                at com.atlassian.event.internal.AsynchronousAbleEventDispatcher.dispatch
(AsynchronousAbleEventDispatcher.java:107)

                at com.atlassian.event.internal.EventPublisherImpl.invokeListeners(EventPublisherImpl.java:160)

                at com.atlassian.event.internal.EventPublisherImpl.publish(EventPublisherImpl.java:79)

                at com.atlassian.plugin.event.impl.DefaultPluginEventManager.broadcast
(DefaultPluginEventManager.java:73)

                at com.atlassian.plugin.manager.DefaultPluginManager.broadcastIgnoreError(DefaultPluginManager.
java:2122)

                at com.atlassian.plugin.manager.DefaultPluginManager.enableDependentPlugins
(DefaultPluginManager.java:1261)

                at com.atlassian.plugin.manager.DefaultPluginManager.addPlugins(DefaultPluginManager.java:1215)

                at com.atlassian.jira.plugin.JiraPluginManager.addPlugins(JiraPluginManager.java:152)

                at com.atlassian.plugin.manager.DefaultPluginManager.earlyStartup(DefaultPluginManager.java:
597)

                at com.atlassian.jira.plugin.JiraPluginManager.earlyStartup(JiraPluginManager.java:120)

                at com.atlassian.jira.ComponentManager$PluginSystem.earlyStartup(ComponentManager.java:641)

                at com.atlassian.jira.ComponentManager.quickStart(ComponentManager.java:195)

                at com.atlassian.jira.ComponentManager.start(ComponentManager.java:164)
```

```
                at com.atlassian.jira.upgrade.PluginSystemLauncher.start(PluginSystemLauncher.java:43)

                at com.atlassian.jira.startup.DefaultJiraLauncher.lambda$postDbLaunch$2(DefaultJiraLauncher.
java:150)

                at com.atlassian.jira.config.database.DatabaseConfigurationManagerImpl.doNowOrEnqueue
(DatabaseConfigurationManagerImpl.java:298)

                at com.atlassian.jira.config.database.DatabaseConfigurationManagerImpl.
doNowOrWhenDatabaseActivated(DatabaseConfigurationManagerImpl.java:194)

                at com.atlassian.jira.startup.DefaultJiraLauncher.postDbLaunch(DefaultJiraLauncher.java:141)

                at com.atlassian.jira.startup.DefaultJiraLauncher.lambda$start$0(DefaultJiraLauncher.java:103)

                at com.atlassian.jira.util.devspeed.JiraDevSpeedTimer.run(JiraDevSpeedTimer.java:31)

                at com.atlassian.jira.startup.DefaultJiraLauncher.start(DefaultJiraLauncher.java:101)

                at com.atlassian.jira.startup.LauncherContextListener.initSlowStuff(LauncherContextListener.
java:149)

                at java.lang.Thread.run(Thread.java:745)

Caused by: com.atlassian.cache.CacheException: java.lang.IllegalArgumentException: Unknown icon type
'issuetype'

                at com.atlassian.cache.memory.DelegatingCachedReference.get(DelegatingCachedReference.java:83)

                at com.atlassian.jira.config.DefaultConstantsManager.getAllIssueTypeObjects
(DefaultConstantsManager.java:636)

                at com.xpandit.raven.service.impl.RavenIssueTypeServiceImpl.findIssueType(Unknown Source)

                at com.xpandit.raven.issuetype.RavenIssueTypeManagerImpl.a(Unknown Source)

                ... 32 more

                at com.atlassian.jira.plugin.icon.IconTypeDefinitionFactoryImpl.getDefaultSystemIconFilename
(IconTypeDefinitionFactoryImpl.java:45)

                at com.atlassian.jira.avatar.AvatarManagerImpl.loadDefaultAvatarId(AvatarManagerImpl.java:488)

                at com.atlassian.jira.avatar.AvatarManagerImpl.lambda$getDefaultAvatarId$3(AvatarManagerImpl.
java:509)

                at com.atlassian.vcache.internal.core.DefaultRequestCache.lambda$get$15(DefaultRequestCache.
java:43)

                at java.util.HashMap.computeIfAbsent(HashMap.java:1126)

                at com.atlassian.vcache.internal.core.DefaultRequestCache.get(DefaultRequestCache.java:43)

                at com.atlassian.jira.avatar.AvatarManagerImpl.getDefaultAvatarId(AvatarManagerImpl.java:509)

                at com.atlassian.jira.avatar.AvatarManagerImpl.getDefaultAvatarId(AvatarManagerImpl.java:482)

                at com.atlassian.jira.issue.issuetype.IssueTypeImpl.getAvatarOrDefault(IssueTypeImpl.java:39)

                at com.atlassian.jira.issue.issuetype.IssueTypeImpl.getAvatar(IssueTypeImpl.java:32)

                at com.atlassian.jira.config.DefaultIssueConstantFactory.createIssueType
(DefaultIssueConstantFactory.java:64)

                at com.atlassian.jira.config.DefaultIssueConstantFactory.createIssueType
(DefaultIssueConstantFactory.java:84)

                at java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:193)

                at java.util.ArrayList$ArrayListSpliterator.forEachRemaining(ArrayList.java:1374)
```

```
                at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:481)

                at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:471)

                at java.util.stream.ForEachOps$ForEachOp.evaluateSequential(ForEachOps.java:151)

                at java.util.stream.ForEachOps$ForEachOp$OfRef.evaluateSequential(ForEachOps.java:174)

                at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)

                at java.util.stream.ReferencePipeline.forEach(ReferencePipeline.java:418)

                at com.atlassian.jira.config.DefaultConstantsManager.loadIssueTypeCache(DefaultConstantsManager.
java:799)

                at com.atlassian.cache.memory.MemoryCacheManager$1$1.load(MemoryCacheManager.java:129)

                at com.atlassian.cache.memory.MemoryCacheManager$1$1.load(MemoryCacheManager.java:105)

                at com.google.common.cache.LocalCache$LoadingValueReference.loadFuture(LocalCache.java:3527)

                at com.google.common.cache.LocalCache$Segment.loadSync(LocalCache.java:2319)

                at com.google.common.cache.LocalCache$Segment.lockedGetOrLoad(LocalCache.java:2282)

                at com.google.common.cache.LocalCache$Segment.get(LocalCache.java:2197)

                at com.google.common.cache.LocalCache.get(LocalCache.java:3937)

                at com.google.common.cache.LocalCache.getOrLoad(LocalCache.java:3941)

                at com.google.common.cache.LocalCache$LocalLoadingCache.get(LocalCache.java:4824)

                at com.google.common.cache.LocalCache$LocalLoadingCache.getUnchecked(LocalCache.java:4830)

                at com.atlassian.cache.memory.DelegatingCachedReference.getUnderLock(DelegatingCachedReference.
java:93)

                at com.atlassian.cache.memory.DelegatingCachedReference.get(DelegatingCachedReference.java:78)

                ... 35 more
```

- Check, querying your database, if any issue type has an invalid avatar id:

```
SELECT * FROM issuetype WHERE avatar NOT IN (SELECT id FROM avatar);
```

- Go to Jira Administration Issues Issue Types;
- Search for the issue types found in the DB query and change its icons to a valid one.
- After performing this changes, restart Jira and the issue should be fixed.

This issue is caused by an inconsistency between two Jira tables: the avatar's table and the issuetype's table. This kind of issue typically happens after an upgrade on Jira.

We are providing a workaround so our customers can still use Xray without having to manually enable it every time they restart Jira.

Project archiving

Can I know the progress of archiving or restoring Xray entities when archiving or restoring the Project?

When a project is archived or restored, Xray kicks in to archive or restore its data (for example its Test Runs). The latter can be a long running task depending on the amount of Xray data to handle.

Currently there is no easy way of knowing the progress of it. But in case you need to know the overall progress, you can refer to the below queries which will report the number of active/archived entities at a given moment.

On Project archive:

PostgreSQL

```
    --The below query returns the number of non-archived test runs in the project. Once it reaches 0, all test runs where successfully archived. Replace <PROJ_KEY> with the project key.
SELECT count(*) as active_test_runs
FROM "AO_8B1069_TEST_RUN" tr
JOIN jiraissue exec_issue on exec_issue.id=tr."TEST_EXEC_ISSUE"
JOIN project p1 on p1.id=exec_issue.project
JOIN jiraissue test_issue on test_issue.id=tr."TEST_ISSUE_ID"
JOIN project p2 on p2.id=test_issue.project
WHERE (tr."ARCHIVED" = 'N' or tr."ARCHIVED" IS NULL)
AND (p1.pkey = '<PROJ_KEY>' OR p2.pkey = '<PROJ_KEY>')

--The below query returns the number of active tests in project test repositories and test plan boards. Once it reaches 0, all tests where successfully hidden. Replace <PROJ_KEY> with the project key.
SELECT count(*) as active_tests
FROM "AO_8B1069_LEAF" leaf
JOIN jiraissue test on test.id=leaf."TEST_ID"
JOIN project p1 on p1.id=test.project
WHERE (leaf."ARCHIVED" = 'N' or leaf."ARCHIVED" IS NULL)
AND (p1.pkey = '<PROJ_KEY>')
```

MySQL

```
    -- The below query returns the number of non-archived test runs in the project. Once it reaches 0, all test runs where successfully archived. Replace <PROJ_KEY> with the project key.
SELECT count(*) as active_test_runs
FROM AO_8B1069_TEST_RUN tr
JOIN jiraissue exec_issue on exec_issue.id=tr.TEST_EXEC_ISSUE
JOIN project p1 on p1.id=exec_issue.project
JOIN jiraissue test_issue on test_issue.id=tr.TEST_ISSUE_ID
JOIN project p2 on p2.id=test_issue.project
WHERE (tr.ARCHIVED = 'N' or tr.ARCHIVED IS NULL)
AND (p1.pkey = '<PROJ_KEY>' OR p2.pkey = '<PROJ_KEY>')

-- The below query returns the number of active tests in project test repositories and test plan boards. Once it reaches 0, all tests where successfully hidden. Replace <PROJ_KEY> with the project key.
SELECT count(*) as active_tests
FROM AO_8B1069_LEAF leaf
JOIN jiraissue test on test.id=leaf.TEST_ID
JOIN project p1 on p1.id=test.project
WHERE (leaf.ARCHIVED = 'N' or leaf.ARCHIVED IS NULL)
AND (p1.pkey = '<PROJ_KEY>')
```

Oracle

```
    --The below query returns the number of non-archived test runs in the project. Once it reaches 0, all test runs where successfully archived. Replace <PROJ_KEY> with the project key.
SELECT count(*) as active_test_runs
FROM "AO_8B1069_TEST_RUN" tr
JOIN jiraissue exec_issue on exec_issue.id=tr."TEST_EXEC_ISSUE"
JOIN project p1 on p1.id=exec_issue.project
JOIN jiraissue test_issue on test_issue.id=tr."TEST_ISSUE_ID"
JOIN project p2 on p2.id=test_issue.project
WHERE (tr."ARCHIVED" = 'N' or tr."ARCHIVED" IS NULL)
AND (p1.pkey = '<PROJ_KEY>' OR p2.pkey = '<PROJ_KEY>')

--The below query returns the number of active tests in project test repositories and test plan boards. Once it reaches 0, all tests where successfully hidden. Replace <PROJ_KEY> with the project key.
```

```
SELECT count(*) as active_tests
FROM "AO_8B1069_LEAF" leaf
JOIN jiraissue test on test.id=leaf."TEST_ID"
JOIN project p1 on p1.id=test.project
WHERE (leaf."ARCHIVED" = 'N' or leaf."ARCHIVED" IS NULL)
AND (p1.pkey = '<PROJ_KEY>')
```

MSSQL

```
    --The below query returns the number of non-archived test runs in the project. Once it reaches 0, all test runs where successfully archived. Replace <PROJ_KEY> with the project key.
SELECT count(*) as active_test_runs
FROM "AO_8B1069_TEST_RUN" tr
JOIN jiraissue exec_issue on exec_issue.id=tr."TEST_EXEC_ISSUE"
JOIN project p1 on p1.id=exec_issue.project
JOIN jiraissue test_issue on test_issue.id=tr."TEST_ISSUE_ID"
JOIN project p2 on p2.id=test_issue.project
WHERE (tr."ARCHIVED" = 'N' or tr."ARCHIVED" IS NULL)
AND (p1.pkey = '<PROJ_KEY>' OR p2.pkey = '<PROJ_KEY>')

--The below query returns the number of active tests in project test repositories and test plan boards. Once it reaches 0, all tests where successfully hidden. Replace <PROJ_KEY> with the project key.
SELECT count(*) as active_tests
FROM "AO_8B1069_LEAF" leaf
JOIN jiraissue test on test.id=leaf."TEST_ID"
JOIN project p1 on p1.id=test.project
WHERE (leaf."ARCHIVED" = 'N' or leaf."ARCHIVED" IS NULL)
AND (p1.pkey = '<PROJ_KEY>')
```

On Project restore:

PostgreSQL

```
    --The below query returns the number of archived test runs in the project. Once it reaches 0, all test runs where successfully restored. Replace <PROJ_KEY> with the project key.
SELECT count(*) as archived_test_runs
FROM "AO_8B1069_TEST_RUN" tr
JOIN jiraissue exec_issue on exec_issue.id=tr."TEST_EXEC_ISSUE"
JOIN project p1 on p1.id=exec_issue.project
JOIN jiraissue test_issue on test_issue.id=tr."TEST_ISSUE_ID"
JOIN project p2 on p2.id=test_issue.project
WHERE tr."ARCHIVED" = 'Y'
AND (p1.pkey = '<PROJ_KEY>' OR p2.pkey = '<PROJ_KEY>')

--The below query returns the number of archived tests in project test repositories and test plan boards. Once it reaches 0, all tests where successfully restored. Replace <PROJ_KEY> with the project key.
SELECT count(*) as archived_tests
FROM "AO_8B1069_LEAF" leaf
JOIN jiraissue test on test.id=leaf."TEST_ID"
JOIN project p1 on p1.id=test.project
WHERE (leaf."ARCHIVED" = 'Y')
AND (p1.pkey = '<PROJ_KEY>')
```

MySQL

```
    -- The below query returns the number of archived test runs in the project. Once it reaches 0, all test runs where successfully restored. Replace <PROJ_KEY> with the project key.
SELECT count(*) as archived_test_runs
FROM AO_8B1069_TEST_RUN tr
JOIN jiraissue exec_issue on exec_issue.id=tr.TEST_EXEC_ISSUE
JOIN project p1 on p1.id=exec_issue.project
JOIN jiraissue test_issue on test_issue.id=tr.TEST_ISSUE_ID
JOIN project p2 on p2.id=test_issue.project
```

```
WHERE tr.ARCHIVED = 'Y'
AND (p1.pkey = '<PROJ_KEY>' OR p2.pkey = '<PROJ_KEY>')

-- The below query returns the number of archived tests in project test repositories and test plan boards. Once
it reaches 0, all tests where successfully restored. Replace <PROJ_KEY> with the project key.
SELECT count(*) as archived_tests
FROM AO_8B1069_LEAF leaf
JOIN jiraissue test on test.id=leaf.TEST_ID
JOIN project p1 on p1.id=test.project
WHERE (leaf.ARCHIVED = 'Y')
AND (p1.pkey = '<PROJ_KEY>')
```

Oracle

```
    --The below query returns the number of archived test runs in the project. Once it reaches 0, all test runs
where successfully restored. Replace <PROJ_KEY> with the project key.
SELECT count(*) as archived_test_runs
FROM "AO_8B1069_TEST_RUN" tr
JOIN jiraissue exec_issue on exec_issue.id=tr."TEST_EXEC_ISSUE"
JOIN project p1 on p1.id=exec_issue.project
JOIN jiraissue test_issue on test_issue.id=tr."TEST_ISSUE_ID"
JOIN project p2 on p2.id=test_issue.project
WHERE tr."ARCHIVED" = 'Y'
AND (p1.pkey = '<PROJ_KEY>' OR p2.pkey = '<PROJ_KEY>')

--The below query returns the number of archived tests in project test repositories and test plan boards. Once
it reaches 0, all tests where successfully restored. Replace <PROJ_KEY> with the project key.
SELECT count(*) as archived_tests
FROM "AO_8B1069_LEAF" leaf
JOIN jiraissue test on test.id=leaf."TEST_ID"
JOIN project p1 on p1.id=test.project
WHERE (leaf."ARCHIVED" = 'Y')
AND (p1.pkey = '<PROJ_KEY>')
```

MSSQL

```
    --The below query returns the number of archived test runs in the project. Once it reaches 0, all test runs
where successfully restored. Replace <PROJ_KEY> with the project key.
SELECT count(*) as archived_test_runs
FROM "AO_8B1069_TEST_RUN" tr
JOIN jiraissue exec_issue on exec_issue.id=tr."TEST_EXEC_ISSUE"
JOIN project p1 on p1.id=exec_issue.project
JOIN jiraissue test_issue on test_issue.id=tr."TEST_ISSUE_ID"
JOIN project p2 on p2.id=test_issue.project
WHERE tr."ARCHIVED" = 'Y'
AND (p1.pkey = '<PROJ_KEY>' OR p2.pkey = '<PROJ_KEY>')

--The below query returns the number of archived tests in project test repositories and test plan boards. Once
it reaches 0, all tests where successfully restored. Replace <PROJ_KEY> with the project key.
SELECT count(*) as archived_tests
FROM "AO_8B1069_LEAF" leaf
JOIN jiraissue test on test.id=leaf."TEST_ID"
JOIN project p1 on p1.id=test.project
WHERE (leaf."ARCHIVED" = 'Y')
AND (p1.pkey = '<PROJ_KEY>')
```

Contact

Can I send you guys an email?

We really prefer that you contact us through our Service Desk, so the right person can properly address your question asap.

I would like a demo. Is that possible?

Sure. Please leave us a message in our Service Desk and we'll get back to you right away. We also have webinars (both live and recorded) that you can attend.

.

.