

Image Artifact viewing tab



Jim Chaney 5 posts since

Oct 23, 2009

One of the artifacts of our current build is a collection of .png images (the 'build' is actually running a complex unit test of some image manipulation functionality). Although I am uploading the images as single artifacts, it is a laborious process to click each link in turn and briefly the images to see if it went 'wrong'.

I looked through the online docs and found two possible avenues:

<http://www.jetbrains.net/confluence/display/TCD4/Including+Third-Party+Reports+in+the+Build+Results>

Using this, I could generate an html page along with the images, zip them all up and add one entry to the main-config.xml. How does this know what build configuration to attach to?

<http://www.jetbrains.net/confluence/display/TCD4/Web+UI+Extensions>

The 'full on' approach, creating an extension that creates the tab dynamically by examining the artifacts and creating the webpage on demand. I'd love to have this (it would be pretty reusable by others) but I am not a web developer and wouldn't really know where to start.

Does anyone have an opinion on whether 1) will work, or even better, opt to write 2) for me and publish it to the JetBrains repository?

Thanks in advance,

Jim

Tags: artifacts, plugin, extension, image, thumbnail

Image Artifact viewing tab



[stefan_b](#) 13 posts since

May 27, 2009 1. **Re: Image Artifact viewing tab** Oct 24, 2009 1:43 PM

Hi Jim,

we use approach 1) which works quite well for us:

The tab-location is relative to the build's artifact dir.

We have something like

```
<report-tab title="Acceptance Tests" basePath="test/acceptance/" />
```

in the main-config.xml and publish an index.html using

```
##teamcity[publishArtifacts 'tmp/acceptancetest/index.html => test/acceptance']
```

This index.html etc. are generated with a ruby script (running on the agent)

and all subtests html-pages and PNGs are pushed to the server via service message, like

```
##teamcity[publishArtifacts 'tmp/acceptancetest/subtest1.html => test/acceptance']
```

```
##teamcity[publishArtifacts 'tmp/acceptancetest/subtest1.png=> test/acceptance']
```

To have a direct link to the test html you could then try to write a plugin using

<http://javadoc.jetbrains.net/teamcity/openapi/current/jetbrains/buildServer/serverSide/FailedTestOutputFormatter.html>

Image Artifact viewing tab

We currently hacked the jsp-page to include our test results, but this not really recommended. The code is below and

is simply added to teamcity/WEBAPPS/root/web-inf/tags/tests/testStacktracePart.tag. This is really a quick hack we did

to migrate from our old system to TC. The java part reads the artifact html, and uses everything inside the body-tags.

The jsp-part simply displays the three standard png-images our regression-tests generate (gold image, generated image, diff image).

Works with 4.5.x version of TC.

Stefan

```
<!-- begin dart tweak -->
```

```
<style type="text/css">
  table.dart_summary {
    visibility: collapse ;
  }
  table.dart {
    color : #ffffff;
    font-family: trebuchet ms, verdana, tahoma, arial, sans-serif;
    font-size:12px;
    border-collapse: collapse;
    padding: 4px;
  }
  .dart td.td-key { border: 1px solid #D6DCE4; padding: 2px; background: #E3E9EF; color:
#0254D0;}
  .dart td.td-value { border: 1px solid #D6DCE4; padding: 2px; background: #FFFFFF; color:
#000040;}
```

Image Artifact viewing tab

```
.dart th.th-key { border: 1px solid #D6DCE4; padding: 2px; background: #E3E9EF; color: #0254D0;}  
.dart th.th-value { border: 1px solid #D6DCE4; padding: 2px; background: #FFFFFF; color: #000040;}  
.dart tr.ruled { background: #cccc00; }  
</style>
```

```
<br/>
```

```
<%
```

```
java.lang.String testType = new java.lang.String();
```

```
if (test.getPackage().endsWith("module"))
```

```
{
```

```
    testType = "module";
```

```
}
```

```
if (test.getPackage().endsWith("acceptance"))
```

```
{
```

```
    testType = "acceptance";
```

```
}
```

```
if (testType.length() > 0)
```

```
{
```

```
    java.io.File myFile = new java.io.File(buildData.getArtifactsDirectory(),"test/" + testType  
+ "/" + test.getPackage() + "/" + test.getShortName()  
+ ".results.xml.test.html");
```

```
    try {
```

```
        java.io.BufferedReader input = new java.io.BufferedReader(new  
java.io.FileReader(myFile));
```

```
        Boolean insideBody = false;
```

```
try {
    String line = null;
    while (( line = input.readLine()) != null)
    {
        if (line.contains("</body>"))
        {
            insideBody = false;
        }
        if (insideBody)
        {
            out.println(line);
        }
        if (line.contains("<body>"))
        {
            insideBody = true;
        }
    }
}
finally {
    input.close();
}
}
%>
```

```
<c:if test="${fn:endsWith(fn:toLowerCase(test.package),'acceptance')}">
<br>
```

```
<a href="repository/download/${buildType.buildTypeId}/${buildData.buildId}:id/test/acceptance/${test.package}/${test.shortName}.diff.png">
```

```

```

```
</a>
```

```
<a href="repository/download/${buildType.buildTypeId}/${buildData.buildId}:id/test/acceptance/${test.package}/${test.shortName}.gold.png">
```

```

```

```
</a>
```

```
<a href="repository/download/${buildType.buildTypeId}/${buildData.buildId}:id/test/acceptance/${test.package}/${test.shortName}.image.png">
```

```

```

```
</a>
```

```
<br>
```

```
</c:if>
```

```
<c:if test="${fn:endsWith(fn:toLowerCase(test.package),'module')}">
```

```
<br>
```

```
<a href="repository/download/${buildType.buildTypeId}/${buildData.buildId}:id/test/module/${test.package}/${test.shortName}.diff.png">
```

```

```

```
</a>
```

```
<a href="repository/download/${buildType.buildTypeId}/${buildData.buildId}:id/test/module/${test.package}/${test.shortName}.gold.png">
```

```

```

```
</a>
```

```
<a href="repository/download/${buildType.buildTypeId}/${buildData.buildId}:id/test/module/${test.package}/${test.shortName}.image.png">
```

Image Artifact viewing tab

```
  
  
</a>  
<br>  
</c:if>  
  
<!-- end dart tweak -->
```



[Jim Chaney](#) 5 posts since

Oct 23, 2009 2. **Re: Image Artifact viewing tab** Oct 26, 2009 1:54 PM

Ok, I think I can see that the code above is going to process and display the tests... but realistically I fall at the first hurdle by not even knowing what markup language it is in! I see the `<c:if />` tag and all of a sudden it isn't normal html anymore...

So, I'll have a go at uploading as a 'third party test report' as the docs describe them... and thanks for your input but I am going to leave this open, see if I can't persuade someone to add something to the default 'Artifacts' tab that spots image files and shows you thumbnails.



[Yegor Yarko](#) 1,459 posts since

May 5, 2004 3. **Re: Image Artifact viewing tab** Oct 26, 2009 3:43 PM

Jim,

Feel free to post a feature request into our [tracker](#). We can then see how many people find this useful by the number of votes and comments.

I am not sure, however, that displaying images as thumbnails in artifacts list is a good approach... For any non-tiny images thumbnail will still needs clicking to see it's content and mixing thumbnails and usual files does not seem usable.

Image Artifact viewing tab

It seems that generating a HTML and displaying it as a report tab is an appropriate approach. For this approach I would consider packing the images into archive.

> How does this know what build configuration to attach to?

The report tab is displayed for every build that has an artifact that matches basePath in artifact tab definition.



[Jim Chaney](#) *5 posts since*

Oct 23, 2009 4. **Re: Image Artifact viewing tab** Oct 29, 2009 3:01 PM

I have had some success in saving xml & xhtml memory usage reports into a zip archive and uploading them, so I suppose I can do the same with the images. I just thought it may be useful to others, to simply have an assets tab that showed images 'inline' so to speak.

Rather than trynig to write JSP, I shall stick with what I know 😊