


# Release Notes 5.2

 [Release Notes 5.1](#)

 [Release Notes](#)

[Release Notes 5.3](#) 

This is the consolidated list of changes between Tapestry versions 5.1 and 5.2. To upgrade from 5.1 to 5.2, most users will be able to just update the Maven dependency in their POM file (or [download](#) the new JAR file) and the new version will just work. However, please read carefully below before upgrading, and also review the [How to Upgrade](#) instructions.

## Contents

- [Breaking Changes](#)
- [Release Notes: Tapestry 5.2.6](#)
- [Release Notes: Tapestry 5.2.5](#)
- [Release Notes: Tapestry 5.2.4](#)
- [Release Notes: Tapestry 5.2.3](#)
- [Release Notes: Tapestry 5.2.2](#)
- [Release Notes: Tapestry 5.2.1](#)
- [Release Notes: Tapestry 5.2.0](#)

## Breaking Changes

The following changes have been made in Tapestry 5.2 that are likely to result in unexpected behavior if your application relies on the changed functionality. Please review this list carefully before upgrading from 5.1 to 5.2. Also check the [Deprecated API List](#) for non-breaking changes.

- Page classes with instance variables that are not thread safe must be created in a method rather than declared as an instance variable. For example, creating an instance variable `private final DateFormat format = DateFormat.getDateInstance(DateFormat.MEDIUM, locale);` in a page and using it will cause problems because `DateFormat` is not thread safe. Instead, you must create the `DateFormat` in a method. See [Release Notes: Tapestry 5.2.0](#) (below) for details.
- [Link.toAbsoluteURI\(\)](#) now returns the absolute URL, which includes the scheme, hostname and possibly port (e.g., "http://example.com:8080/myapp/viewproduct/4"), rather than a relative URL (e.g., "/myapp/viewproduct/4"). See [Release Notes: Tapestry 5.2.2](#) (below) for details.
- The [Label](#) component no longer outputs an id:

Previously valid code in 5.1.0.5:

```
<t:form><t:label for="search"/><t:textfield t:id="search" size="50"/></t:form>
```

Combined with JavaScript that references the id:

```
jQuery('#search-label').text('This is the search label');
```

New JavaScript code for 5.2:

```
jQuery('label[for=search]').text('This is the search label');
```

- `ContributeTranslatorSource` compatibility is broken. Service 'TranslatorSource' is now configured using a `MappedConfiguration`, not a `Configuration`. See issue [TAP5-1395](#)
- Code that uses `Form.getDefaultTracker().getHasErrors()` fails if it assumes that `getDefaultTracker()` is not null when validation is successful. See issue [TAPS-1401](#).
- Virtual folders, used to define root packages for component libraries, may no longer contain slashes.
- The path prefix of a `LibraryMapping` may no longer contain the slash character.

## Release Notes: Tapestry 5.2.6

Tapestry 5.2.6 is a maintenance release.

## Bugs Fixed

- [\[TAP5-1368\]](#) - The `onFailure` function provided to `Tapestry.ajaxRequest` does not always get invoked
- [\[TAP5-1546\]](#) - `TranslatorSource` incorrectly annotated with `@UsesConfiguration` when it should now be `@UsesMappedConfiguration` (as of 5.2)
- [\[TAP5-1551\]](#) - `FormFragment` change visibility/hide and remove event listeners should call `event.stop()` to prevent container fragments from also being hidden

## Improvements Made

- [\[TAP5-1482\]](#) - Upgrade Prototype to 1.7 / Scriptaculous 1.9.0

- [\[TAP5-1538\]](#) - Optimize client-side URL rebuilding in IE 7
- [\[TAP5-1539\]](#) - Optimize document scans used by Tapestry.FieldEventManager to not locate the label or icon until actually needed

## Release Notes: Tapestry 5.2.5

Version 5.2.5 is a maintenance release for 5.2.4.

### Bugs Fixed

- [TAP5-1372](#) - BaseURLSource uses getLocalPort() rather than getServerPort()

### Improvements Made

- [TAP5-1040](#) - Allow HTML5 doctype to be printed
- [TAP5-1208](#) - In development mode, Tapestry should "shadow" field & parameter values to instance variables, to assist with debugging
- [TAP5-1346](#) - Fix broken links on new Web Site

## Release Notes: Tapestry 5.2.4

Version 5.2.4 replaces version [5.2.3](#), which had an incorrect version number in the quickstart archetype.

### Improvements Made

- [TAP5-1394](#) – The method parameter annotations are missing from Tapestry IoC proxies

## Release Notes: Tapestry 5.2.3

### Bugs Fixed

- [TAP5-1204](#) – tapestry-core JAR includes some compiled Groovy test scripts
- [TAP5-1326](#) – Bug in ConcurrentBarrierTest
- [TAP5-1327](#) – Tapestry IOC depends on SLF4J 1.5.6 which is not compatible with Hibernate 3.6 (which depends on 1.6.1)
- [TAP5-1328](#) – Tapestry.ajaxRequest() fails when just parameters are provided, but not an onSuccess handler function
- [TAP5-1330](#) – Position of Injected Stylesheets changed from 5.1 to 5.2
- [TAP5-1332](#) – In a multi-zone update, if the referenced Zone can not be found on the client, Tapestry stops updating Zones with no error
- [TAP5-1334](#) – Id attribute of the @Decorate annotation should be optional
- [TAP5-1345](#) – A supported locale name is only recognized in lower case

### Improvements Made

- [TAP5-929](#) – Implement sendRedirect() for TestableResponseImpl
- [TAP5-1333](#) – Zone should implement an interface, BodyElement, that extends ClientElement and provides the body property
- [TAP5-1341](#) – Improve page testing facilities
- [TAP5-1342](#) – Provide convenient methods for Element and Document to find elements

## Release Notes: Tapestry 5.2.2

It is now possible to generate an absolute URL from a Link object; an absolute URL includes the URL scheme, host, port number (if necessary) followed by the complete path. This is useful in a number of situations, such as when generating a URL that may be e-mailed to an end user.

The semantics of the `Link.toAbsoluteURL()` method has changed; in prior releases it returned a complete URL (i.e., starting with a leading slash), and the method was typically used when generating an Ajax callback URL. Starting in 5.2.2 this generates an absolute URL (as described above). In most cases, you should change code that invoked `toAbsoluteURL()` with a call to `toURL()` instead; because Tapestry no longer 'optimizes' URLs, this will work for Ajax cases as well.

Deployed Tapestry applications often run behind a firewall, the available Request information about incoming hostname and port is often not accurate (it represents the private request from the firewall server to the application server, whereas the client uses a public request to the firewall server). When using absolute URLs, you will likely want to override the BaseURLSource service to provide accurate values that reflect your application's deployment.

### Bugs Fixed

- [TAP5-708](#) – Element.forceAttribute uses the element's namespace to match the attribute.
- [TAP5-709](#) – Element.attribute(String name, String value) adds elements that already exist
- [TAP5-804](#) – Element#addClassName can create an additional new 'class' attribute
- [TAP5-921](#) – Properties defined in an Interface are not exposed by PropertyAccess for abstract classes that do not directly implement the methods
- [TAP5-1287](#) – Some services require a notification that they have been reloaded, so they can clean up external dependencies
- [TAP5-1293](#) – Whitespaces in SymbolConstants.SUPPORTED\_LOCALES cause that locales are not persised
- [TAP5-1294](#) – Validation macros do not work when used in @Validate annotation
- [TAP5-1297](#) – Client-side validation of @Pattern is broken

- [TAP5-1303](#) – Linking a Form to a Zone will no longer work unless the Form contains validated fields
- [TAP5-1304](#) – When using PropertyShadowBuilder to build a service, if the property is null, an immediate exception is needed (rather than a NullPointerException)
- [TAP5-1309](#) – When using a MultiZoneUpdate, Tapestry will clear the referenced zone

## Improvements Made

- [TAP5-538](#) – Add the facility to optionally disable on-focus-change triggered validation while retaining the on-form-submit validation
- [TAP5-874](#) – Form component should be able to render a secure URL even on an insecure page
- [TAP5-1272](#) – New annotations @Decorate and @Advise to identify methods that decorate or annotate services
- [TAP5-1291](#) – Extend Link with new methods for producing absolute URLs (that include scheme, hostname, etc.)
- [TAP5-1295](#) – Simplify connecting a link or form to an (enclosing) Zone
- [TAP5-1298](#) – BeanBlockContribution should be split into two sub-classes: EditBlockContribution and DisplayBlockContribution
- [TAP5-1300](#) – Define a special CSS class to prevent a client-side form from submitting normally (for Ajax use cases)
- [TAP5-1306](#) – Additional method for Link: addParameterValue(String, Object) that uses ContextPathEncoder to encode object value to a string
- [TAP5-1314](#) – SeleniumTestCase should expose the underlying CommandProcessor, to allow execution of commands not defined by the Selenium interface
- [TAP5-1315](#) – Allow individual SeleniumTestCases to run w/o configuring SeleniumLauncher

## New Features Added

- [TAP5-1299](#) – Option to disable live service reloading via a JVM system property

## Tasks Completed

- [TAP5-1317](#) – Upgrade Hibernate dependency to latest, Hibernate 3.6.0.Final

## Release Notes: Tapestry 5.2.1

Prior releases of Tapestry attempted to "optimize" URLs, by creating relative URLs where appropriate. Unfortunately, that caused its own problems with Ajax requests, since the base URL for an Ajax request is not the same as the base URL for the enclosing page. In any case, this feature has been removed, all URLs are now complete URLs (they start with a leading slash, and include the context path and the additional path that describes the page, page activation context, etc.).

As part of this change, the *internal* RequestPathOptimizer service has been removed.

## Bugs Fixed

- [TAP5-710](#) – Documentation about "use of asset: and context: to reference assets directly from a template file" has errors (See TAP5-214)
- [TAP5-752](#) – Tapestry bin download includes lots of libraries that shouldn't be there due to licensing concerns
- [TAP5-770](#) – Minor documentation typos in PageLoaded annotation
- [TAP5-818](#) – Tapestry should properly support JDK 1.5 Generics when reading and updating properties and property expressions
- [TAP5-1004](#) – X-Tapestry-ErrorMessage may lead to HTTP Response Splitting
- [TAP5-1011](#) – When using @PageActivationContext and providing a no-args activate event handler, the handler may be called too soon
- [TAP5-1118](#) – Using Ajax updates to add an Upload component to an existing Form fails in IE, which still sends application/x-www-form-urlencoded encoded request
- [TAP5-1123](#) – Documentation in ApplicationStateManager is not quite correct: when creating an SSO it uses the full injection machinery, not necessarily the no-args constructor
- [TAP5-1177](#) – In a partial page update request, if the server returns an empty string for the content, the client-side is left as is, rather than cleared
- [TAP5-1188](#) – Refactoring the base class of a live-reloaded service implementation can cause mysterious NoSuchMethodError exceptions
- [TAP5-1192](#) – IE 8 throws an exception while purging JavaScript event handlers from elements as they are destroyed
- [TAP5-1201](#) – PageResponseRenderer.render() should call RequestGlobabls.storeActivePageName(String pageName)
- [TAP5-1222](#) – Accessing a public field of a non-component object inside component code can result in a TransformationException if the accessed field name matches the name of a component field
- [TAP5-1228](#) – The need to configure the tapestry.alias-mode symbol makes it awkward to start a Registry up for unit testing purposes
- [TAP5-1229](#) – Tapestry binary/source distributions do not include all modules
- [TAP5-1230](#) – SeleniumTestCase.assertFieldValue() should invoke writeErrorReport() on failure
- [TAP5-1232](#) – Restore injection of service id (as a String) into service decorator methods for proper backwards compatibility
- [TAP5-1235](#) – The interaction between a Form component and the active page can be problematic when there is an error loading a page
- [TAP5-1241](#) – Setting the content length of the response should automatically disable response compression
- [TAP5-1244](#) – Tapestry components and mixins should be converted to use JavaScriptSupport, not RenderSupport
- [TAP5-1246](#) – Tapestry should use SQLException.getNextException() to dig down to the "root cause"
- [TAP5-1254](#) – The decorateComponentEventLink event name is misspelled
- [TAP5-1258](#) – Maven quickstart archetype puts the wrong version in the pom.xml
- [TAP5-1260](#) – Document "Assets", Section "Securing Assets" is outdated
- [TAP5-1261](#) – Tapestry adds event handlers not only to form components, but also to simple forms
- [TAP5-1264](#) – Ordering of MarkupRenderer (and PartialMarkupRender) filters can cause bad interaction between ClientBehaviorSupport and Heartbeat
- [TAP5-1265](#) – Change the description for the quickstart archetype to just "Tapestry 5 Quickstart Project"
- [TAP5-1269](#) – Annotation @FactoryDefaults should be allowed on methods, for use with the @Contribute annotation
- [TAP5-1277](#) – Tapestry incorrectly calls PrintWriter.flush(), not close(), when writing JSON objects and arrays as the response
- [TAP5-1279](#) – Importing a JavaScript library that is part of a stack should import the entire stack instead of the individual library (to take advantage of JavaScript aggregation)

## Improvements Made

- [TAP5-363](#) – Cookbook: Using Tapestry to serve servlet-container error pages
- [TAP5-1233](#) – Live reloading of service implementation classes causes `IllegalAccessError` if the class contains protected or package private methods
- [TAP5-1236](#) – Add method `Response.disableCompression()` (to avoid use of a internal global key)
- [TAP5-1245](#) – Break out the Tapestry JSON support into an independent module
- [TAP5-1250](#) – Add support for creating a Flow from an Iterable (not just Collection and object array)
- [TAP5-1256](#) – Failed Selenium assertions should capture a screenshot (as well as the current page HTML)
- [TAP5-1266](#) – Bring quickstart archetype up to date in several ways

## New Features

- [TAP5-910](#) – Cookbook: defining a new enum as a component parameter, and adding the string -> enum type coercion for it

## Tasks Completed

- [TAP5-1240](#) – Upgrade Hibernate dependencies to 3.5.4-Final
- [TAP5-1267](#) – Remove the code that "optimizes" generated URLs as it is not always possible to know what the base URL is
- [TAP5-1270](#) – Upgrade EasyMock dependency to version 3.0

## Release Notes: Tapestry 5.2.0

Tapestry 5.2.0 represents nearly a year of work past the prior stable release, 5.1.0.5. For the most part, the upgrade is quite straight forward, but please read the notes below carefully.

### Live Service Reloading

Tapestry 5.2.0 extends Tapestry's concept of live reloading of Java code into the service layer. In most cases, service implementations of your application will now live reload, seamlessly. See the [FAQ](#) for some additional notes about live reloading.

### Pages No Longer Pooled

This is a huge change from Tapestry 5.1 to 5.2; Tapestry no longer pools page instances. It creates **one** instance of each page (per supported locale). Tapestry rewrites your component classes so that all mutable state is stored in a per-thread `HashMap`. This will greatly reduce Tapestry's memory footprint, especially on heavily loaded sites.

This change makes it easier to share objects between threads, which is problematic when the objects are not thread-safe. For example:

#### Valid Component Code (5.1)

```
@Inject
private Locale locale;

private final DateFormat format = DateFormat.getDateInstance(DateFormat.MEDIUM, locale);

public String getCurrentTimeFormatted()
{
    return format.format(new Date());
}
```

In the above code, *under 5.1*, the `DateFormat` object was not shared between threads, as each thread would operate with a different instance of the containing page. Under 5.2, the final field *will* be shared across threads, which is problematic as `DateFormat` is not thread safe. The code should be rewritten as:

## Updated for 5.2

```
@Inject
private Locale locale;

public String getCurrentTimeFormatted()
{
    DateFormat format = DateFormat.getDateInstance(DateFormat.MEDIUM, locale);
    tapestry.page-pool-enabled
    return format.format(new Date());
}
```

If such issues are difficult to solve, remember that as a temporary work-around you have the option of re-enabling the page pooling behavior of 5.1 by setting `tapestry.page-pool-enabled` to "true" in your [configuration](#).

## Service Id Injection

In prior releases of Tapestry, a constructor parameter of type `String` was assumed to be the service id. In the many cases where these was not the case (such as using the `@Value` or `@Symbol` annotation), the parameter needed to be annotated with the `@Inject` annotation.

Starting in 5.2.0, this feature is no longer present (this is a *non-backwards compatible change*, but one that affects virtually nobody). A parameter of type `String` will be subject to normal injection; you will likely want to use `@Value` or `@Symbol` with it, or you will see an error that "No service implements java.lang.String."



Igor made a change such that a bare `String` is injected as the service id in some circumstance; it may be decorator methods. This needs to be documented properly.

## TranslatorSource

The configuration type for `TranslatorSource` has changed in an *incompatible* way: from an unordered collection to a mapped collection; this is to support overrides of Tapestry's built-in translators. This will break existing module classes that contribute to the `TranslatorSource` service configuration.

## Assets

There have been some changes to how assets operate in Tapestry 5.2.

Virtual folders, used to define root packages for component libraries, may no longer contain slashes. Virtual folders are the `pathPrefix` property of the `LibraryMapping` objects that are contributed to the `ComponentClassResolver` service.

Each `LibraryMapping` is now automatically converted into a matching contribution to the `ClasspathAssetAliasManager` service. Previously a library author was encouraged to make contributions to both services. The path prefix of a `LibraryMapping` is also now prohibited from containing the slash character.

It is now quite necessary to configure the application version number: all assets are exposed via a URL that incorporates the application version number; in previous releases, each library could configure its own version number. By implication, changing library versions and nothing else will now require a change to the application version number.

## ClassTransformation API changes

The `ClassTransformation` API, used to implement component meta-programming, has been rewritten with an eye to making class transformations easier to implement while removing the dependency on the Javassist library. This is of note only to *advanced* users who have implemented `ComponentClassTransformWorkers`, which operate on the `ClassTransformation` API.

Much of the existing API has been deprecated and *some* deprecated methods are now non-functional. The deprecated methods will be removed in Tapestry 5.3.

This represents a shift in how the API operates; previously much of the logic in `ComponentClassTransformWorkers` was built in terms of Javassist: adding new bytecode to existing methods. The new API switches away from this, and now works in terms of adding new fields, initializing those fields using callbacks, providing callbacks for access to fields, and adding advice to methods.

## Template Parser back to SAX

Tapestry no longer uses a StAX parser to parse component templates, it has reverted to using a normal SAX parser. This change reduces the number of dependencies for Tapestry, and is a stepping stone towards compatibility with Google App Engine.

## Bugs Fixed

- [TAP5-266](#) – In a conflict between a render phase annotation and the naming convention, the explicit annotation should win
- [TAP5-707](#) – Yellow highlight remains on updated zone if zone is re-updated too quickly
- [TAP5-711](#) – Submit component: using image parameter prevents selected event from being fired
- [TAP5-714](#) – Incorrect encoding of single quotes for Ajax requests
- [TAP5-715](#) – TypeCoercer.explain incorrectly reports the plan to coerce from primitive types to wrapper types
- [TAP5-719](#) – Component LinkSubmit doesn't work
- [TAP5-728](#) – When using the @Validate annotation, spaces around the commas that separate constraints cause runtime exceptions
- [TAP5-734](#) – Tapestry tutorial documentation refers to old archetype command
- [TAP5-747](#) – Property expressions that include method invocations that in turn reference properties result in spurious error about "root"
- [TAP5-748](#) – NPE when defining a component using just t:id and no type or @Component annotation
- [TAP5-749](#) – The FormFragment and LinkSubmit components create a hidden field whose id ends with ".hidden"
- [TAP5-750](#) – Tapestry should not attempt to GZip flash movies (.swf files)
- [TAP5-755](#) – URL rewriting documentation contains an example that won't compile due to lack of a return value
- [TAP5-759](#) – The documentation of DefaultNullFieldStrategy#replaceFromClient() is wrong about what gets returned
- [TAP5-765](#) – Included JavaScript libraries are not properly uniqued within an Ajax partial update response
- [TAP5-767](#) – PropertyConduitSourceImpl should use english locale (instead of default locale) when evaluating decimals
- [TAP5-769](#) – JavaScript aggregation can be inefficient across multiple pages with different JS requirements
- [TAP5-774](#) – Initialization JavaScript needs help when order counts
- [TAP5-779](#) – CLONE -Linksubmit doesn't work inside a form with Zone parameter set
- [TAP5-786](#) – Exception messages in the pageload packages should be localized
- [TAP5-787](#) – AbstractIntegrationTest's "assertTextPresent" only checks the first value
- [TAP5-788](#) – ParallelExecutor service throws RejectedExecutionException if more than THREAD\_POOL\_MAX\_SIZE jobs are submitted
- [TAP5-796](#) – Form component generates invalid xHtml: there should not be the "name" attribute
- [TAP5-812](#) – The input validation documentation incorrectly shows validation occurring in the success event handler method
- [TAP5-813](#) – the component rendering page of the user guide is not generated by maven
- [TAP5-815](#) – Asset dispatcher allows any file inside the webapp visible and downloadable
- [TAP5-823](#) – Message about incompatible return type of a render phase method is misleading
- [TAP5-824](#) – Javassist 3.9.0.GA not available from central repository
- [TAP5-830](#) – SpringModuleDef unnecessarily hard-codes that the type of object stored in the context as a ConfigurableWebApplicationContext when any object implementing ApplicationContext is sufficient
- [TAP5-834](#) – BaseOptimizedSessionPersistedObject does not work correctly with Tomcat & Jetty
- [TAP5-837](#) – prototype1.6.0 not support for ie8
- [TAP5-839](#) – Tapestry should ignore (public) synthetic methods in module classes
- [TAP5-856](#) – MetadataLocatorImpl.findMeta(String, String, Class) doesn't check contributed defaults - breaks SECURE\_PAGE contributions
- [TAP5-868](#) – It is not possible to attach a validation event listener to a Palette (or other <select> field)
- [TAP5-871](#) – Generation of Component Reference failed on Windows
- [TAP5-881](#) – Tapestry's customized Blackbird implementation sends a dubious cookie value over every request
- [TAP5-894](#) – Fix PartialMarkupDocumentLinkerTest.stylesheet\_link()
- [TAP5-896](#) – Contribute 'properties' file extension to the configuration of ResourceDigestGenerator
- [TAP5-898](#) – BeanModel methods reorder, include and exclude should return BeanModel<T> (not BeanModel<?>)
- [TAP5-908](#) – Blackbird console should not add cookies to requests
- [TAP5-913](#) – Stack size too large exception, related to PropertyExpressionLexer
- [TAP5-919](#) – Calling MarkupWriter.element() with an even number of arguments throws ArrayIndexOutOfBoundsException exception, not a useful exception about omitting a attribute name or value
- [TAP5-923](#) – Injecting Tapestry Services to Spring Beans is completely undocumented
- [TAP5-936](#) – Tapestry wiki link links to nothing for locales other than en
- [TAP5-937](#) – LinkImpl does not handle parameters properly when passed into the constructor
- [TAP5-940](#) – Zone should fire a heart beat event
- [TAP5-943](#) – Documentation for event handler methods should go into detail about returns true or false
- [TAP5-944](#) – When a ValueEncoder is unable to convert an id to a entity, it should wrap the underlying type coercion exception to describe the input and expected output type
- [TAP5-945](#) – Unnecessary and severe lock contention in PerThreadManagerImpl
- [TAP5-947](#) – Default name "submit" for submit component breaks javascript function submit()
- [TAP5-959](#) – Ajax event handler methods that return a page instance, page class or page name should cause the client to redirect to that page, but doesn't work in IE7 or 8 if the URL has a query param
- [TAP5-961](#) – When a URL rewriting rule changes the path for a rendered link (component event or page render), any query parameters associated with the original link are lost
- [TAP5-962](#) – Errors component includes an @Environmental for FormSupport, but doesn't use it
- [TAP5-964](#) – Exception report failed to render due to NPE inside session-persisted objects' toString()
- [TAP5-974](#) – PageCallback should be marked with the @ImmutableSessionPersistedObject annotation
- [TAP5-987](#) – In some cases you can invoke Form.recordError() and the Form will still fire a success (not a failure) event
- [TAP5-1001](#) – Garbled exception message when a component is defined in a namespace that is not properly defined as a Tapestry library namespace
- [TAP5-1018](#) – Request to Application Root path ignores ComponentRequestFilter's
- [TAP5-1019](#) – French translations for number format error message
- [TAP5-1020](#) – Fix typo in tapestry-hibernate/src/site/apt/userguide.apt: HibernateTransactionAdviser should be HibernateTransactionAdvisor
- [TAP5-1030](#) – Injecting a Logger into a non-service object, created as part of constructing a service configuration, should work but doesn't
- [TAP5-1031](#) – NPE from Any component when invoking getClientId() before the component renders
- [TAP5-1034](#) – Using URLRewriteRules will fail for component event links if the page has an activation context
- [TAP5-1042](#) – URLRewriting causes confusion about the incoming Request path, leading to incorrect optimized relative URLs
- [TAP5-1043](#) – PropertyDisplayBlocks causes NullPointerException when property of type Calendar is null
- [TAP5-1047](#) – @IncludeJavaScriptLibrary and @IncludeStylesheet may incorrectly calculate resources in component sub-classes
- [TAP5-1052](#) – Component classes page of the guide should be much more specific that only component classes go in pages, components, etc.
- [TAP5-1054](#) – Not able to pass an empty string as appName to PageTester
- [TAP5-1057](#) – XSS vulnerability in calendar component
- [TAP5-1060](#) – In IE8, forms that should submit using Ajax are triggering full page requests due to a client-side JavaScript error
- [TAP5-1067](#) – Created component constructor may use too many parameters
- [TAP5-1068](#) – RenderSupport.addScriptLink(String) deprecated and removed without replacement
- [TAP5-1069](#) – Tapestry POM should reference only the Apache Nexus repository, not the tapestry.formos.com repository



- [TAP5-1074](#) – The `@QueryParam` annotation uses `TypeCoercer` to convert from request parameter strings to method parameter values, it should use the `ValueEncoder`
- [TAP5-1075](#) – When a template contains a reference to an unknown component, the new `UnknownValueException` is reported but the `Location` is no longer reported, making it much harder to determine where the error occurred
- [TAP5-1076](#) – When a service implementation is reloadable, it will not eager load
- [TAP5-1080](#) – Page activation context lost when redirecting from HTTP to HTTPS due to the `@Secure` annotation
- [TAP5-1086](#) – Localized root path request results in a 404 error if Index page is not available
- [TAP5-1088](#) – Bean editor model for `<class name>` already contains a property model for property `<property name>` when combining model parameter with add parameter of `BeanEditor` or `BeanEditForm`
- [TAP5-1090](#) – Deleting a live-reloaded class can result in a non-descriptive "Parameter url was null." exception
- [TAP5-1096](#) – If a Zone's id parameter is bound, then the `clientId` property should return that value, even before the Zone renders
- [TAP5-1100](#) – On ajax request, template body doesn't render if page is new in pagepool or if server restarted
- [TAP5-1105](#) – `BeanModelSource` should recognize public field as properties, but doesn't
- [TAP5-1106](#) – `Tapestry.ajaxFailureHandler` has an incorrect signature for an `Ajax.Request` on `Exception` callback
- [TAP5-1109](#) – Updating multiple zones within a Form creates anomalous empty text fields
- [TAP5-1110](#) – `Tapestry` holds onto the verbose descriptions of component class transformations, causing a memory leak
- [TAP5-1116](#) – Live lock when using compressed and virtual assets due to `ByteArrayOutputStream` being synchronized
- [TAP5-1120](#) – It is not possible to override the default `Translator` contributions to the `TranslatorSource` service
- [TAP5-1124](#) – `FormEventManager.setSubmittingElement(element)` creates input element with `type="input"`
- [TAP5-1128](#) – A space is added to a HTML start comment sequence making it impossible to add IE conditional comments
- [TAP5-1130](#) – `LinkSubmit` component doesn't work when form contains `Select` component with default id
- [TAP5-1136](#) – `java.lang.UnsupportedOperationException` from `UpdateListenerHub` when a weak reference is reclaimed
- [TAP5-1137](#) – Dynamically adding JavaScript libraries to a page via a partial page update does not seem to work consistently in Safari and Chrome
- [TAP5-1138](#) – `ClassTransformation` deprecated a bunch of methods, then refers to replacements that don't exist
- [TAP5-1141](#) – Type coercion from arbitrary object to `Boolean` will NPE when the object's `toString()` returns null (should coerce to false)
- [TAP5-1145](#) – Writing only text (or raw text) into a `Document` via `MarkupWriter` results in [empty Document], not the actual text
- [TAP5-1147](#) – Ajax updates under IE leak lots of memory
- [TAP5-1154](#) – `ValidationDecorator.insideField` not called for `Select` component
- [TAP5-1156](#) – Provide a `ValueEncoder` for a Hibernate entity automatically only if a mapped class exists
- [TAP5-1158](#) – Captured HTML from failed Selenium assertions do not get stored in the correct place on Windows
- [TAP5-1161](#) – `Tapestry 5.1` seems incompatible with `Spring 3`
- [TAP5-1166](#) – `ClassPropertyAdapterImpl` should skip static fields
- [TAP5-1182](#) – `javax.management` prevents `Tapestry` apps from being deployed on GAE
- [TAP5-1185](#) – Upgrade to latest `Javassist` version to avoid `LocalVariableTable` exceptions (especially on Google App Engine)
- [TAP5-1198](#) – A user defined activate event handler appears to execute "before" the event handler supplied by `@PageActivationContext`, making defensive coding impossible
- [TAP5-1203](#) – Use of `@Contribute` annotation does not work properly with marker annotations
- [TAP5-1207](#) – A form control component (such as `TextField`) whose id is "id" can confuse client-side logic for the Form DOM object
- [TAP5-1209](#) – Marker annotations used with services that have contributions should be applicable to methods (as well as fields and parameters) for compatibility with `@Contribute`
- [TAP5-1210](#) – `MultiZone` update from `EventLink` with Form fails in Internet Explorer
- [TAP5-1211](#) – `LinkSubmit` broken in IE 7 mode
- [TAP5-1216](#) – `ValueEncoderSource` does not define a `@UsesMappedConfiguration` annotation
- [TAP5-1223](#) – The name "JavaScript" should have a capital "S" but is inconsistent in some class and interface names
- [TAP5-1224](#) – When contributing to a service using a "contribute" method, the service id from the method name now matches case-sensitively to the service id (it should be case insensitive)
- [TAP5-1262](#) – XSS vulnerability in calendar component (apply to 5.1.0.x)
- [TAP5-1282](#) – Form component generates invalid xHtml: there should not be the "name" attribute - Applied to 5.1

## Improvements Made

- [TAP5-69](#) – Add annotation, `@Contribute`, to allow service contributor methods to be arbitrary named
- [TAP5-88](#) – Add support for Bulgarian locale
- [TAP5-179](#) – The `TriggerFragment` mixin would be more useful if it could invert its logic, i.e., make a a fragment visible when a checkbox was turned off
- [TAP5-247](#) – Move all `Tapestry 5` configuration keys to a constant class
- [TAP5-335](#) – Provide access to annotations of service implementation class
- [TAP5-424](#) – Allow component libraries to contribute extra resources to the global application catalog
- [TAP5-632](#) – Property names (in property expressions) should be able to read or update public variables
- [TAP5-674](#) – Make it easy to implement a page callback mechanism
- [TAP5-678](#) – Allow blackbird to be disabled in production mode
- [TAP5-685](#) – Allow services defined by module `tapestry-ioc` to be advised and decorated
- [TAP5-713](#) – Change template parser to not use `StAX`, as it is not (yet) compatible with Google App Engine
- [TAP5-760](#) – The Form event "validateForm" is awkwardly named and should be replaced with the simpler name "validate"
- [TAP5-762](#) – Upgrade Selenium dependencies to version 1.0.1
- [TAP5-764](#) – Hidden should support `ClientElement` and support informal parameters.
- [TAP5-777](#) – `Tapestry` should ensure that mixins are applied in a deterministic order.
- [TAP5-783](#) – `tapestry` should mark via annotation which annotations are component-specific vs. more general
- [TAP5-789](#) – Provide bean and display blocks for `java.util.Calendar`
- [TAP5-790](#) – Provide `ApplicationStatePersistenceStrategy` for Hibernate entities
- [TAP5-803](#) – `ProgressiveDisplay` should include a read-only body property of type `Block`, just like `Zone`
- [TAP5-807](#) – `PageRenderLinkSource` should add additional methods for creating a `Link` when you have the page's activation context as an `EventContext`
- [TAP5-810](#) – Improve documentation of `@Persist` annotation
- [TAP5-811](#) – Layout component documentation is not very clear about directory structure
- [TAP5-814](#) – include a diagram showing how incoming requests pass through the different pipelines, filters and dispatchers
- [TAP5-826](#) – The `ActionLink` documentation is missing the option to use object arrays as context inside templates

- [TAP5-859](#) – In a stack trace, any lines associated with line #1 of a class are likely to be synthetic methods and should be classified as "uninteresting"
- [TAP5-863](#) – Tapestry.ajaxRequest is insufficient when doing extra, such as adding query parameters
- [TAP5-884](#) – Introduce a symbol for tapestry.js
- [TAP5-887](#) – Easier way to initialize a JSONObject with keys and values
- [TAP5-889](#) – Provide fluent API for order constraints of contributions
- [TAP5-904](#) – Make use of Selenium.showContextualBanner() when running integration tests
- [TAP5-905](#) – Tapestry should support the full range of Unicode characters acceptable by Java as property names
- [TAP5-912](#) – Validation of properties of type java.util.Collection should fail when the collection is empty
- [TAP5-914](#) – When autobuilding a Java bean (such as a Session State Object), the code should use the OperationTracker
- [TAP5-915](#) – It should be possible to override a components message catalog
- [TAP5-917](#) – Don't set Expires header when tapestry.production-mode is false
- [TAP5-927](#) – Cannot use Scala for Tapestry IOC Modules
- [TAP5-948](#) – Built-in mechanism to identify self-referential links and/or event/render requests
- [TAP5-958](#) – Upgrade Tapestry's built-in copy of prototype.js to version 1.6.1
- [TAP5-963](#) – Allow access to static resources (css, js, jpg, jpeg, png, gif) inside the app package
- [TAP5-965](#) – Upgrade EasyMock dependency to release 2.5.2
- [TAP5-967](#) – ObjectLocator.autobuild would be more useful with an override that allowed a message about the object to be described
- [TAP5-971](#) – FormFragment component should include a parameter to control whether non-visible content is included in the form submission
- [TAP5-978](#) – Provide remote management of the page pool settings
- [TAP5-979](#) – Form component should be more careful with the validation tracker to ensure that a session is not created unless needed
- [TAP5-993](#) – Reorganize ComponentClassTransformWorkers to start moving away from Javassist
- [TAP5-994](#) – Field autofocus should be done via Tapestry.init instead of using \$ function
- [TAP5-1000](#) – When Autobuilding a class, the constructor to be used should be identified in the trace output
- [TAP5-1014](#) – Message about missing or wrong retention policy of a marker annotation is misleading
- [TAP5-1015](#) – Provide a new return type for event handler methods that would trigger the rendering of a particular page without a redirect
- [TAP5-1023](#) – (5.2.0-SPNAPSHOT Maven dependencies) tapestry-hibernate-5.2.0 jar generated with tapestry-hibernate-5.2.0-yyyyMMdd.hhhmmss-?.jar > tapestry-hibernate-5.2.0-SNAPSHOT.jar whereas all other jar are using -5.2.0-SNAPSHOT
- [TAP5-1024](#) – Submit component should allow return values from event handler methods triggered by its event
- [TAP5-1033](#) – Upgrade selenium dependencies to 1.0.3
- [TAP5-1036](#) – It would be nice if there was a way for committers to run individual integration tests again
- [TAP5-1037](#) – Rewrite URLRewriter integration tests to use new SeleniumTestCase instead of deprecated AbstractIntegrationTestSuite
- [TAP5-1038](#) – Rewrite live reload integration tests to use new SeleniumTestCase instead of deprecated AbstractIntegrationTestSuite
- [TAP5-1046](#) – Change Tapestry client-side JavaScript to make the tapx/Confirm component easier to implement
- [TAP5-1050](#) – Add API to prevent a particular service implementation from being live reloaded
- [TAP5-1055](#) – Provide hook to post-process properties files before rolling them into component Messages
- [TAP5-1061](#) – When a Zone component sends an Ajax request for a client-side update, it should pass an extra query parameter identifying the zone's client-side id
- [TAP5-1077](#) – Merge symbol values into property values from a bean definition
- [TAP5-1085](#) – The ExceptionReporter should be configured to mark more of the Tapestry generated classes (used for method & field access and method advice) as hidden by default
- [TAP5-1091](#) – CLONE -Handler method of LinkSubmit component should accept a context
- [TAP5-1094](#) – Create a binding prefix, "symbol:", that is used to access IoC symbols
- [TAP5-1112](#) – Handle array types in property expressions
- [TAP5-1126](#) – Add a new validator, "none", used when overriding the @Validate annotation
- [TAP5-1127](#) – Documentation for the Submit and LinkSubmit components should identify that you should cancel defer when inside a Loop
- [TAP5-1129](#) – LinkSubmit should render minimally (and as <span> tag) and then fill in the details on the client
- [TAP5-1132](#) – HibernateGridDataSource should assign the result of Projections.rowCount() to java.lang.Number when determining the number of rows for the configured entity type
- [TAP5-1135](#) – Provide a convenient method to build and start the registry
- [TAP5-1150](#) – Refactor out a new base class between org.testng.Assert and org.apache.tapestry5.ioc.test.TestBase for non-mock related tests
- [TAP5-1152](#) – TypeCoercer should include a method exposing the coercion it would use for a given source and target type
- [TAP5-1153](#) – When in development mode, Tapestry should pretty-print JSON content
- [TAP5-1155](#) – JavaScript initialization inside the partial page render Ajax response should be unquoted
- [TAP5-1157](#) – Client-side exceptions during Tapestry.init() should be caught and reported with the console
- [TAP5-1165](#) – Make better use of the OperationTracker to identify what's going on during a request (and especially, during page construction)
- [TAP5-1170](#) – Remove automatic injection of service id into parameters of type String
- [TAP5-1174](#) – Provide a ComponentEventResultProcessor that sends an error response to the client
- [TAP5-1197](#) – Eliminate page pooling using shared page instances that separate their structure from the mutable state
- [TAP5-1225](#) – Use recent Groovy/GMaven version and fix GMaven configuration in tapestry-core POM

## New Features

- [TAP5-52](#) – Add Error component that presents validation errors of a single field
- [TAP5-56](#) – Tapestry should have support for IE conditional stylesheets
- [TAP5-86](#) – Add support for "cancel" submit buttons (which bypass client-side validation)
- [TAP5-103](#) – provide access to component parameters from within mixins
- [TAP5-138](#) – Add Zone parameter to Select component
- [TAP5-152](#) – Add @Translate annotation to define name of translator to be used with a bean property (rather than lookup by property type)
- [TAP5-156](#) – Add a @QueryParameter annotation for parameters to event handler method
- [TAP5-226](#) – Add annotation @SessionAttribute to map a field to a specific session attribute
- [TAP5-680](#) – Tapestry should provide a mixin for ensuring that the client id of a ClientElement is rendered
- [TAP5-692](#) – T5 should pick up validators to be applied to a field from the containing component's .properties file.
- [TAP5-801](#) – Add Trigger component to trigger a component event during rendering
- [TAP5-895](#) – Tracking issue for Tapestry/JSR-303 integration
- [TAP5-951](#) – Create more flexible API for testing with Selenium that allows for multiple test cases to share a single instance of Selenium /SeleniumServer/Jetty
- [TAP5-966](#) – TapestryFilter should be able add additional modules to the Registry to accomodate different testing (or other) execution configurations



- [TAP5-1013](#) – Live class reloading for service implementations
- [TAP5-1028](#) – Validator Macros: Combine multiple common validators into a single term
- [TAP5-1035](#) – In places where an invalid key is used to accessed a named value, Tapestry should report the possible names better (using HTML lists, rather than a long comma-separated string)
- [TAP5-1056](#) – The application global message catalog should be injectable into services
- [TAP5-1064](#) – Extend PropertyAccess to understand Scala style properties (which use a different naming convention than JavaBeans)
- [TAP5-1065](#) – Non-visual mixin to generate events as it begins/ends rendering (useful for generating matching JavaScript)
- [TAP5-1079](#) – Live class reloading should extend to proxied objects (such as from ObjectLocator.proxy())
- [TAP5-1084](#) – Zones that initially render inside a Form should support updates within the Form
- [TAP5-1095](#) – LocalizationSetter service should expose the supported locales as a list of Locale and as a SelectModel
- [TAP5-1097](#) – New annotation: @HeartbeatDeferred to mark component methods that should execute at the end of the current Heartbeat
- [TAP5-1098](#) – Provide new SelectModelFactory service that can automatically build a standard SelectModel from objects and property names
- [TAP5-1099](#) – Introduce public service responsible for handling page activation
- [TAP5-1121](#) – Provide an annotation to support automatic discarding of the persistent fields after a component or page method invocation
- [TAP5-1159](#) – Easy way to customize search locations for page and component templates
- [TAP5-1190](#) – New page-level events to "decorate" component event and page render links

## Tasks Completed

- [TAP5-11](#) – CookiesImplTest does specify a domain cookie with a domain not prefixed with a . (dot)
- [TAP5-556](#) – Fix TranslatorSourceImplTest
- [TAP5-756](#) – Add ioko-tapestry-commons to the related projects list
- [TAP5-819](#) – remove ide-specific files from all sub-modules and add them to svn:ignore
- [TAP5-969](#) – Method AbstractField.createDefaultParameterBinding() should be deprecated
- [TAP5-976](#) – Upgrade Spring dependencies to version 3.0.0.RELEASE
- [TAP5-1081](#) – Remove formos references from 5.2.0 archetype
- [TAP5-1087](#) – Upgrade TestNG dependencies to version 5.12.1
- [TAP5-1134](#) – Upgrade Hibernate dependencies to 3.5.2
- [TAP5-1195](#) – Rename annotations @QueryParameter and @QueryParameterMapped (both introduced in 5.2.0) to more mnemonic names