

Jubula Release Notes

Jubula Team

BREDEX GmbH

June 29, 2012



BREDEX GmbH
Mauernstr. 33
38100 Braunschweig
Germany
Tel: +49-531 - 243 30 - 0
Fax: +49-531 - 243 30 - 99
www.bredex.de

GUI*dancer* is a registered trademark of BREDEX GmbH

Title: Jubula Release Notes
Author: Jubula Team
File: releaseNotes
State: RELEASE
Version: V6.0.01011
Released by: BREDEX GmbH
Released at: June 29, 2012

Jubula Release Notes

This document presents the relevant differences between versions and updates, and provides an account of any developments or known issues with the current release.

For up-to-date information on a release, it is worth looking in the FAQ's on the Jubula website.

The release notes are presented in chronological order, with the most recent at the beginning of the document.

Important advice for migrating to new Jubula versions

Existing customers who wish to update to a new version of Jubula should follow the steps described in the user manual to ensure a problem-free migration.

Release Notes for Jubula V6.0.01011

New Features and Developments

Chronon support

- Jubula now allows the use of an embedded Chronon recorder and also the production of Chronon recording files during automated tests (Jubula users: this feature is currently only available in the standalone version).
- Activating the embedded Chronon recorder in Jubula will result in your actions being recorded in a monitoring file which can be used for analysis purposes should you encounter any errors.
- You can also configure your AUT to use Chronon as a monitoring agent so that you can generate recording files from your automated tests.
- The currently used version of Chronon is 2.0. There are some performance issues still in this version that may lead to test runs and performance being affected, particularly in larger AUT's and Projects. Please follow the instructions in the documentation for increasing heap size and ensuring the capability of your test machines.

- We do not recommend having Chronon running at all times, due to possible performance degradations through the monitoring. Instead, we recommend that you activate the Chronon recorder when you are trying to reproduce an error.

Tests updated to newer versions of IE and Firefox

- The tested versions for web AUT's now include IE 9 and Firefox 10

Known issues and other information

Documentation error in Chronon section

- The documentation for using Chronon while running an AUT specifies that the package pattern must be a regular expression. This is incorrect. The correct text should be:

Enter a comma-separated list of packages that you want to be covered by the monitoring. The packages must adhere to the patterns as defined in the Chronon documentation, for example `com.myorg.**` selects the whole `com.myorg` namespace. `com.myorg.*` selects only classes in the `com.myorg` package. The documentation for the patterns is located at <https://chronon.onconfluence.com/display/DOC/Include+and+Exclude+patterns>. If you enter no patterns, the recording file will be empty.

Chronon and continuous integration

Although you could technically use Chronon with Jubula outside of the ITE in Continuous Integration environments, neither BREDEX nor Chronon Systems recommends or will provide support for such cases.

The current integration with Chronon is meant to be used inside of the ITE. If you do want to record your tests outside of the ITE, please look at Chronon Systems' Chronon recording server:

<http://www.chrononsystems.com/products/chronon-recording-server/>

and Embedded Chronon:

<http://www.chrononsystems.com/products/embedded-chronon> offerings which are explicitly designed for and support such use cases.

Release Notes for Jubula version 6.0007x

New Features and Developments

Test Execution Duration now shown for each node in Test Result View

- The amount of time taken to execute each Test Step, Test Case and Test Suite is now shown in the Test Result View.
- You can use this information to see whether your tests or parts of your tests are taking longer than you expected them to, without having to set timers or manually check timestamps.
- The HTML and XML reports generated or exported for the test results also show the execution times.
- You can switch off the decoration in the Jubula client in the *label decoration* preferences.

Parameter values now shown for each node in Test Result View

- The parameter values entered or used at a specific level (Test Case, Test Step) are now shown in the Test Result View.
- You can use this information to see which data were used for your Test Cases, without having to click through each individual node in the Test Result View. This can be especially useful if you have one Test Case that runs multiple times with different datasets.
- You can switch off the decoration in the Jubula client in the *label decoration* preferences.

Missing data decoration now shown in the Test Case Editor and Test Suite Editor

- When working on a Test Case or Test Suite in their respective editors, you will now see small red crosses on any referenced Test Cases or Test Steps in that editor which have missing data *at this level* (i.e. data that should be entered in this editor).

- This will help avoid problems with forgetting to enter data in the Properties View.

Functions can be entered as parameter values

- Alongside concrete values (abc), references (=P1) and variables (\$VAR), you can now enter functions as data, or as parts of your data.
- The sign to introduce a function is the question mark: ?



Any question marks used as test data will have to be escaped using one or more backslashes as of this new version. A warning in the console view will appear after importing your Project if you have any unescaped question marks in your data. You can then use the search function to find them and mask them.

- There are functions available out of the box for various mathematical calculations and for working with dates.
- Functions can be embedded in other functions, e.g. ?add(?sub(3,2)), to add the result of (3-2) to 4.
- Functions can also make use of other data entry options in Jubula such as variables (?add(\$VAR1,=P1) adds the number in the variable VAR1 to the number in the reference P1).
- You can add your own functions via an extension point.

Categories now supported in the Test Suite Browser and Central Test Data Editor

- It is now possible to create categories for Test Suites and Test Jobs in the Test Suite Browser, as well as for central data sets in the Central Test Data Editor.
- Existing Projects will be shown as having two categories (*Test Suites* and *Test Jobs*) in the Test Suite Browser as was previously the case. However, you can now rename, delete and rearrange categories in the Test Suite Browser.

Teststyle framework and some guidelines now also in Jubula

- There is a new properties page in the Project properties for Teststyle.
- The Teststyle framework is a feature that allows Projects to be analyzed for their adherence to standards, best practices and rules for test specification.
- The framework was previously a part of **GUIDancer** and has now been migrated to Jubula.
- Jubula contains a small set of standard rules for test specification. Teststyle in **GUIDancer** contains more guidelines and rules for naming conventions, test specification and test structure.

Analysis framework in Jubula

- A new analysis framework has been added to Jubula to gain information on the size, ratios and details of a Project.
- The analysis framework currently contains three example metrics that can be run on existing or new Projects.

Save As New Test Case

- There is a new option in the Test Case Editor and Test Suite Editor to save selected items as a new Test Case.
- The selected nodes are added to a new Test Case whose name you define.
- The nodes are not copied, but their references are: the effect is the same as if you had manually created a new Test Case and added the Test Cases to it.

New action: Store Property Value

- The *abstract* toolkit contains a new action on the *Graphics Component* component to store the value of a property.
- The action works in a similar way to the *Check Property* action, except that the expected value must not be entered. Instead, the actual value is saved into a variable you define.
- You can use this value later on in the test, or you can compare various values using the actions to compare string values or compare numeric values.

- One example use case for this new action is to test table filters. The rowCount of the table can be saved into a variable before filtering, and the rowCount after filtering into a different variable. The variables can then be compared with each other to ensure that the second value is less than the first, for example.

New extensible adapter mechanism for Swing components

- Jubula now allows you to add support for renderers for Swing components without the getText() method.
- An example of the adapter mechanism can be found here: <http://git.eclipse.org/c/jubula/org.eclipse.jubula.core.git/tree/org.eclipse.jubula.examples.extension.swing.rc.adapter>
- This does not replace the support for custom Swing renderers that can be changed directly by your developers.
- If you are able to change the renderers yourself, you can still implement one of the following in your renderer:

```
public String getTestableText(); public String getText();
```

Completeness check shown as progress window

- After saving in the ITE, the completeness check which validates the state of the Project (test data, object mapping, missing Test Cases etc) is now shown explicitly as a progress window.
- This results from performance improvements made to the completeness check.

Eclipse Marketplace Client now accessible from within Jubula

- You can now open the Eclipse Marketplace Client from the **Help** menu.
- You can use the Marketplace Client to search for new software to install.
- Please be aware that we do not test Jubula for compatibility with plugins installed via the Marketplace that are not provided by BREDEX GmbH or the Eclipse Jubula project and therefore recommend caution when installing new software.

Changes to layout of properties view for test results

- The properties view for test results now shows all test data as two columns: the parameter name and the value.
- This reduces the amount of space required to see test data for test results.

GD_AUT_STARTUP_DELAY renamed to TEST_AUT_STARTUP_DELAY

- There is an undocumented variable that can be set as an environment variable to increase the delay between starting an AUT and checking that it is there.
- This delay was named GD_AUT_STARTUP_DELAY and has now been updated to TEST_AUT_STARTUP_DELAY. Any customers using this variable should update it in their test environment.

Selenium update

- The version of Selenium used by Jubula has been updated to 2.12.0.

Known issues and other information

Text input problems on Mac systems for RCP (Indigo)

- Due to an issue in SWT, text input of characters involving the »SHIFT« key (upper case letters, some special characters) on Mac for RCP AUT's that use Indigo as a target platform does not work in this version.
- The issue has been fixed in SWT and will be available in Eclipse Juno.

New workspaces required

- Due to changes to internal IDs, a new workspace is required for this version.
- You can export your database preferences from your old workspace and import them into your new one.

- Other preferences must be noted and manually updated in the new workspace.

AUT Agent takes longer to start

- In the current version, the AUT Agent takes longer to start than in previous versions.
- If you try to connect to an AUT Agent before it is running, you will receive an information message that the connection could not be established.
- You may need to bear this in mind for automated tests in a continuous build and test environment.

Release Notes for Jubula version 5.2.00265

New Features and Developments

Embedded gdagent

- If you are starting your AUT and running your tests on your local machine, you can now connect to an embedded AUT Agent directly from the ITE.
- This saves you having to start an AUT Agent on localhost.
- This is also useful for testers working with Jubula as a feature in an Eclipse installation.
- The embedded AUT Agent is started on port 60000 by default; this number can be changed in the preferences.

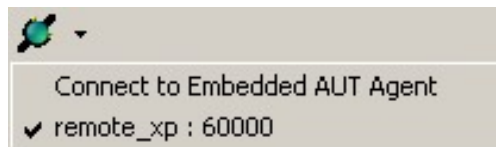


Figure 1: *Embedded AUT Agent*

Refactor: Replace with Test Case

- In the Test Case Editor and Test Suite Editor, there is a new option to replace one or more selected Test Cases with another Test Case from the library.

- A new wizard takes you step-by-step through the replacement process, letting you transfer component names, match parameters and add further information for the new Test Case.
- This feature should help testers who want to restructure their tests after creating a reusable module to replace one or more Test Cases.

Object Mapping Editor: Cleanup unused component names

- In the Object Mapping Editor, there is a new option in the context-sensitive menu.
- Via **Cleanup** → **unused component names** you can start a search for any component names that are no longer used in Test Suites for this AUT.
- Once the search is finished, you can delete all of these unused names from the Object Mapping Editor. If they are then no longer used in the entire Project, they can be deleted from the Component Name Browser.



Figure 2: Deleting unused Component Names

HTML Test Result Reports can be expanded again

- Following changes made for the initial contribution to Eclipse, the HTML Test Result Reports could not be viewed properly in the previous version.

- In this release, the nodes in the HTML reports can be expanded and collapsed to view the whole test progress.

Test Result Report (Complete)

Execution Information	
Project Name	MultiToolkit 5.2
Start Time	Thu Jul 21 23:26:41 CEST 2011
End Time	Thu Jul 21 23:26:52 CEST 2011
Duration	00:00:10
Expected Test Steps	19
Executed Test Steps	19
Event Handler Test Steps	0
Failed Test Steps	0
Language	Deutsch (Deutschland)
Test Suite Information	
Name	MultiToolkit-FULLTEST-SWT
Status	SUCCESSFUL
Application Under Test	
Name	SWT_Adder
Configuration	hudson_win@localhost
Hostname	localhost
AUT Arguments	
Execution Stack	
MultiToolkit-FULLTEST-SWT - TC (ok)	
AdderTestWithTestData - TC (ok)	
complex_app_init - TC (ok)	
AdderTest - TC (ok)	
AdderTest - TC (ok)	
AdderTest - TC (ok)	

Figure 3: HTML Reports

State of component recognition displayed when collecting technical names

- When a component (technical name) is collected from an AUT, it receives a colored dot corresponding to the accuracy of the object recognition for this component *at the time of collecting*.

A green dot signifies that the component could be found with an exact match, and was the only component above the threshold.

A yellow dot means that the component is an exact match, but that multiple other components were also above the threshold.

A red dot means that this component cannot be relocated in the current state of the AUT

- As a side effect, the colors on the component names (green) and technical names (red) that were displayed in previous

versions are now no longer shown. Once the Object Mapping Editor has been saved, all names are shown with plain black icons.

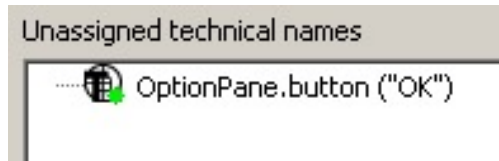


Figure 4: Colored Dots for Object Mapping

Migration wizard re-enabled

- When migrating to the new version of Jubula, the migration assistant will automatically notify you that your database scheme is out-of-date.
- You can then select which Projects to import (these must have been exported from the database prior to migrating!).
- The assistant will drop the tables in the database, recreate the necessary tables and import the Projects you specified.



Figure 5: Migration Wizard

Copy ID to Clipboard also for Test Suites

- The ability to copy a unique ID to the clipboard to reference a Jubula element in external systems has also been implemented for Test Suites.
- You can now copy the ID of a Test Case or a Test Suite to the clipboard, and also open an element based on its ID in the clipboard using »SHIFT+F9«

The DBTool is more verbose

- The command line tool to import, export and delete Projects in the database, the DBTool, has been updated so that it is more verbose.

Progress View

- The ITE now uses the progress view to show longer-running activities such as test execution, connecting to AUT's etc.

Edit Parameters Dialog in Central Test Data Editor can be opened via double-click

- In the Central Test Data Editor it is no longer necessary to open the Edit Parameters Dialog via context-menu, as it can now also be opened via double-click on the data set you wish to edit.

Mac keyboards now supported, new mechanism for adding keyboard layout files

- In the AUT configuration dialog for SWT and RCP AUT's, the *Keyboard Layout* combo box now only offers layouts that have been defined for Jubula. German (DE) and English (US) are defined as standard.
- As well as being able to add keyboard layouts for other keyboards, you can also define platform-specific keyboard layouts (e.g. for Mac)
- The documentation has been updated to describe the new mechanism for adding keyboard layouts.

Known issues and other information

autrun directory has changed

- The *autrun* application has been moved from its own directory in the installed version to within the *server* folder. The *autrun* directory no longer exists.
- Scripts using the *autrun* option will have to be adjusted to reflect this change.

The type of error an Event Handler reacts to cannot be changed once the Event Handler has been added

- In the current version, the type of error you choose for an Event Handler cannot be changed in the Properties View once the Event Handler has been added.
- If you want to change the error type for an Event Handler, then you must delete it and re-add it with the correct error type.

JFace decoration error

- Under some circumstances, there may be an error in the decoration resulting from our use of the `org.eclipse.jface.resource.CompositeImageDescriptor`. This results in a Null Pointer exception.
- If this occurs, please restart Jubula.

Necessary SWT Version

- To be able to test AUT's on Mac systems, the required SWT version is 3.6.
- The required SWT version for Windows and Unix systems is 3.1.

French (FR) keyboards

- The current keyboard layout mechanism for SWT and RCP keyboards does not work for French (FR) keyboards.
- Testers with AUT's in French should consider using a Canadian keyboard layout or any other keyboard where the base keys do not include special characters such as ampersand.