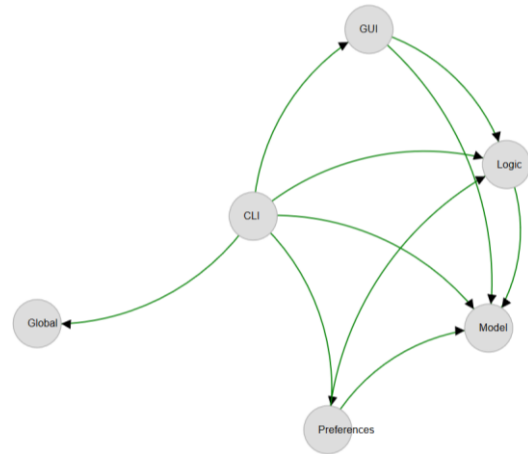


JabRef Architecture

A - Module Architecture

	model	logic	prefs	gui	cli	global
model	+1	+1	+1	+1	+1	-1
logic	-1	+1	+1	+1	+1	-1
prefs	-1	-1	+1	-1	+1	-1
gui	-1	-1	-1	+1	+1	-1
cli	-1	-1	-1	-1	+1	-1
global	-1	-1	-1	-1	+1	+1



B - Module Descriptions

<p>Model: "The <u>model</u> module is the <u>inner core</u> of JabRef with its most important classes. It contains the <u>data representations</u> for the objects that JabRef is working on, in particular its most important <u>structures</u> such as <u>bib databases</u>, including <u>entries</u>, <u>metadata</u> and <u>groups</u>. Basically all <u>data representation classes</u> should go into this package, so the module also includes representations for <u>key patterns and events</u>. Because of its role as the <u>core</u>, classes in the package are not allowed to depend on other JabRef classes and they should be largely independent of particular technologies and programming frameworks"</p>	<p><u>model inner core data representation structure bib database, entries, metadata group key pattern event</u> role core not allowed to depend independent technologies programming frameworks</p>
<p>Logic: "This is arguably the second <u>most important module</u> in JabRef. It should <u>contain functions that operate on the classes in the model package</u>. So the module is not about <u>data representation</u>, but about <u>data manipulation</u>. The key point here is that the functions should be <u>independent of UI technologies</u>, so that we can <u>reuse them easily with different technologies</u>. This means that very many different functions are included in this module, in particular things like <u>backup</u> and "<u>file monitoring</u>", parsing of <u>aux files</u>, computation of <u>key patterns</u>, <u>citations styles</u>, <u>import</u> and <u>export</u>, <u>layouting</u> and <u>formatting</u>, <u>groups-related functions</u>, <u>help</u>, <u>integrity checking</u>, <u>journal lists</u>, <u>localization</u>, <u>logging</u>, <u>support</u> for <u>Open/Libre Office</u> and <u>Word</u>, <u>pdf</u> and <u>xmp processing</u>, <u>search</u> and more. Like the <u>model</u> package, the <u>logic</u> package should be <u>independent from other packages</u> and <u>only depend on the data classes in the model package</u></p>	<p><u>data manipulation independent UI technologies reuse different technologies backup file monitor, parsing aux file, computation key pattern, citation style, import export, layout format, group, help, integrity checking, journal list, localization, log, support Open/Libre Office Word, pdf xmp processing, search logic independent depend data</u></p>
<p>Preferences: "JabRef manages a lot of <u>user configuration options</u>. These are captured in a big and several smaller <u>preferences</u> classes. The preferences module should be independent of any <u>other program</u>, since it is just about <u>storing key-value pairs</u>. The <u>gui</u> module should read the preferences during runtime and pass appropriate options to the logic module as parameters"</p>	<p><u>user configuration option preference independent storing key-value pair runtime pass option parameter</u></p>
<p>GUI: "This is the biggest and most messy part of JabRef. Here, <u>anything goes</u> and any dependencies are allowed. The <u>gui</u> layers should use classes from all other layers to weave the UI of JabRef</p>	<p><u>anything any dependencies gui weave UI together UI element gui representation document viewer, entry editor field editor,</u></p>

<p>together. There is a certain correspondence to the structure of the logic module, since the functionality provided there is being used by dedicated UI elements. Apart from gui representations for the functionality, the core parts of the gui module are the document viewer, the entry editor with field editors, keyboard and desktop classes for OSX, Linux, and Windows, the main table, menus, the base panel and the frame"</p>	<p>keyboard desktop OSX, Linux, Windows, main table, menu, base panel frame</p>
<p>CLI: The cli module is very small and responsible for running JabRef and accessing its functionality without a UI. It parses command line arguments and maps them on certain functions from the logic module. However, the feature set here is much smaller"</p>	<p>cli access without UI parse command line argument maps feature set much smaller</p>
<p>Global: This is no actual module or package, but it bundles the root classes in JabRef, in particular the main class, the main GUI class, a Globals class for storing a limited set of global values. The uniform exception model and fallback handler and finally an executor service for background tasks. The point of these classes is to tie everything together for an orderly startup and an orderly shutdown</p>	<p>bundle root main GUI Globals storing limited global values uniform exception model fallback handler executor service background task tie everything together startup shutdown</p>

C – Class-Module Mappings

1. GUI

- 1.1. net.sf.jabref.gui.*
- 1.2. net.sf.jabref.pdfimport.*
- 1.3. net.sf.jabref.collab.Change
- 1.4. net.sf.jabref.collab.ChangeDisplayDialog
- 1.5. net.sf.jabref.collab.ChangeScanner
- 1.6. net.sf.jabref.collab.EntryAddChange
- 1.7. net.sf.jabref.collab.EntryChange
- 1.8. net.sf.jabref.collab.EntryDeleteChange
- 1.9. net.sf.jabref.collab.FileUpdatePanel
- 1.10. net.sf.jabref.collab.GroupChange
- 1.11. net.sf.jabref.collab.InfoPane
- 1.12. net.sf.jabref.collab.MetadataChange
- 1.13. net.sf.jabref.collab.PreambleChange
- 1.14. net.sf.jabref.collab.StringAddChange
- 1.15. net.sf.jabref.collab.StringChange
- 1.16. net.sf.jabref.collab.StringNameChange
- 1.17. net.sf.jabref.collab.StringRemoveChange
- 1.18. net.sf.jabref.migrations.FileLinksUpgradeWarning
- 1.19. net.sf.jabref.JabRefGUI
- 1.20. net.sf.jabref.JabRefMain

2. CLI

- 2.1. net.sf.jabref.cli.*

3. Logic

- 3.1. net.sf.jabref.logic.*
- 3.2. net.sf.jabref.collab.FileUpdateListener
- 3.3. net.sf.jabref.collab.FileUpdateMonitor
- 3.4. net.sf.jabref.shared.DBMSProcessor
- 3.5. net.sf.jabref.shared.DBMSSynchronizer

- 3.6. net.sf.jabref.shared.MySQLProcessor
- 3.7. net.sf.jabref.shared.OracleProcessor
- 3.8. net.sf.jabref.shared.PostgreSQLProcessor
- 3.9. net.sf.jabref.shared.event.*
- 3.10. net.sf.jabref.shared.exception.*
- 3.11. net.sf.jabref.shared.listener.*
- 3.12. net.sf.jabref.JabRefExecutorService

4. Global

- 4.1. net.sf.jabref.Globals

5. Model

- 5.1. net.sf.jabref.model.*
- 5.2. net.sf.jabref.shared.security.*
- 5.3. net.sf.jabref.shared.DBMSConnection
- 5.4. net.sf.jabref.shared.DBMSType
- 5.5. net.sf.jabref.JabRefException

6. Preferences

- 6.1. net.sf.jabref.preferences.*
- 6.2. net.sf.jabref.migrations.PreferencesMigrations
- 6.3. net.sf.jabref.shared.DBMSConnectionProperties
- 6.4. net.sf.jabref.shared.prefs.*

???? - External

- net.sf.jabref.FallbackExceptionHandler

Generated classes not mapped e.g.

- net.sf.jabref.search.SearchBaseVisitor
- net.sf.jabref.search.SearchLexer
- net.sf.jabref.search.SearchParser
- net.sf.jabref.search.SearchVisitor