**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**

|  |  |
| --- | --- |
| **Model:** ~~"The~~ **model** module ~~is the~~ inne**r core** ~~of~~ JabRef ~~with its~~ most important classes. It contains the data **representations** for the objects tha**t** JabRef is working on, in particular its most important **structures** such as **bib databases**, including **entries**, **metadata**~~and~~ **groups**. Basically all data representation classes should go into this package, so the module also includes representations ~~for~~ ke**y patterns** ~~and~~ **events**. Because ~~of its~~ role ~~as the~~ cor**e,** 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" | **model** inner **core** data **representation** **structure** **bib database**, **entries**, **metadata group** key **pattern** **event** role corenot allowed to depend independent technologies programming frameworks |
| **Logic:** "This is arguably the second most important module in JabRef. It should contain functions that operate on the classes in the model package. So the module is not about data representation, but about data **manipulation**. The key point here is that the functions should be independent of UI technologies, so that we canreuse ~~them easily with~~ different technologies. This means that very many different functions are included in this module, in particular things like **backup** and ”**file monitoring”**, parsing~~of~~ **aux files**, computation of **key patterns**, **citations styles**, **import** and **export**, **layouting** and **formatting**, **groups**-related functions, **help**, **integrity checking**, **journal lists**, **localization**, **logging**, **support** for **Open/Libre Office** and **Word**, **pdf** and **xmp** processing, **search** and more. Like the model package, the **logic** package should be independent from other packages and only depend on the data classes in the model package | 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 |
| **Preferences:** "JabRef manages a lot of user **configuration options**. These are captured in a big and several smaller **preferences** classes. The preferences module should be independent of any other program, since it is just about **storing key-value** pairs. The gui module should read the preferences during runtime and pass appropriate options to the logic module as parameters" | user **configuration option** **preference** independent **storing key-value** pair runtime pass option parameter |
| **GUI:** "This is the biggest and most messy part of JabRef. Here, anything goes and any dependencies are allowed. The **gui** layers should use classes from all other layers to weave the UI of JabRef 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**" | anything any dependencies **gui** weave UI together **UI element** gui representation **document viewer**, **entry editor** **field editor**, **keyboard desktop** **OSX**, **Linux**, **Windows**, **main table**, **menu**, **base panel** **frame** |
| **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" | **cli** access without UI **parse command line argument** maps feature set much smaller |
| **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** | bundle **root** **main GUI** **Globals** storing limited global values uniform **exception model** **fallback handler** **executor service background task** tie everything together **startup** **shutdown** |

**C – Class-Module Mappings**

1. **GUI**
   1. net.sf.jabref.gui.\*
   2. net.sf.jabref.pdfimport.\*
   3. net.sf.jabref.collab.Change
   4. net.sf.jabref.collab.ChangeDisplayDialog
   5. net.sf.jabref.collab.ChangeScanner
   6. net.sf.jabref.collab.EntryAddChange
   7. net.sf.jabref.collab.EntryChange
   8. net.sf.jabref.collab.EntryDeleteChange
   9. net.sf.jabref.collab.FileUpdatePanel
   10. net.sf.jabref.collab.GroupChange
   11. net.sf.jabref.collab.InfoPane
   12. net.sf.jabref.collab.MetaDataChange
   13. net.sf.jabref.collab.PreambleChange
   14. net.sf.jabref.collab.StringAddChange
   15. net.sf.jabref.collab.StringChange
   16. net.sf.jabref.collab.StringNameChange
   17. net.sf.jabref.collab.StringRemoveChange
   18. net.sf.jabref.migrations.FileLinksUpgradeWarning
   19. net.sf.jabref.JabRefGUI
   20. net.sf.jabref.JabRefMain
2. **CLI**
   1. net.sf.jabref.cli.\*
3. **Logic**
   1. net.sf.jabref.logic.\*
   2. net.sf.jabref.collab.FileUpdateListener
   3. net.sf.jabref.collab.FileUpdateMonitor
   4. net.sf.jabref.shared.DBMSProcessor
   5. net.sf.jabref.shared.DBMSSynchronizer
   6. net.sf.jabref.shared.MySQLProcessor
   7. net.sf.jabref.shared.OracleProcessor
   8. net.sf.jabref.shared.PostgreSQLProcessor
   9. net.sf.jabref.shared.event.\*
   10. net.sf.jabref.shared.exception.\*
   11. net.sf.jabref.shared.listener.\*
   12. net.sf.jabref.JabRefExecutorService
4. **Global**
   1. net.sf.jabref.Globals
5. **Model**
   1. net.sf.jabref.model.\*
   2. net.sf.jabref.shared.security.\*
   3. net.sf.jabref.shared.DBMSConnection
   4. net.sf.jabref.shared.DBMSType
   5. net.sf.jabref.JabRefException
6. **Preferences**
   1. net.sf.jabref.preferences.\*
   2. net.sf.jabref.migrations.PreferencesMigrations
   3. net.sf.jabref.shared.DBMSConnectionProperties
   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