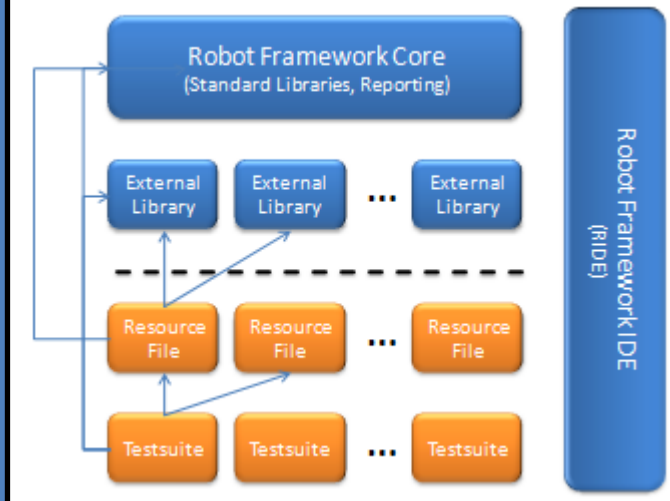
Robot Framework – *Compact Sheet*

codecentric

Abstract: The *Robot Ramework* is a generic Test Automation Framework for acceptance testing. It has a flexible keyword-driven approach to describe and implement tests.

Keywords can be stored in Resource-Files to separate implementation details from tests. Main programming languages for implementing test functionality are Java and Python.

There are ready-made test libraries available for a lot of technologies. Implementing tests is supported by the Robot Framework IDE(RIDE).



Available Test Libraries

- BuildIn
- Dialogs
- Collections
- OperatingSystem
- Remote
- Screenshot
- String
- Telnet
- XML
- Android Library
- Anywhere Library
- Appium Library
- Archive Library
- AutoIt Library
- Database Library
- Diff Library
- Eclipse Library
- robotframework-faker
- FTP Library
- HTTP Library
- iOS Library
- MongoDB Library
- Rammbock
- RemoteSwingLibrary
- SeleniumLibrary
- Selenium2Library
- SSH Library
- Suds Library
- Swing Library
- watir-robot

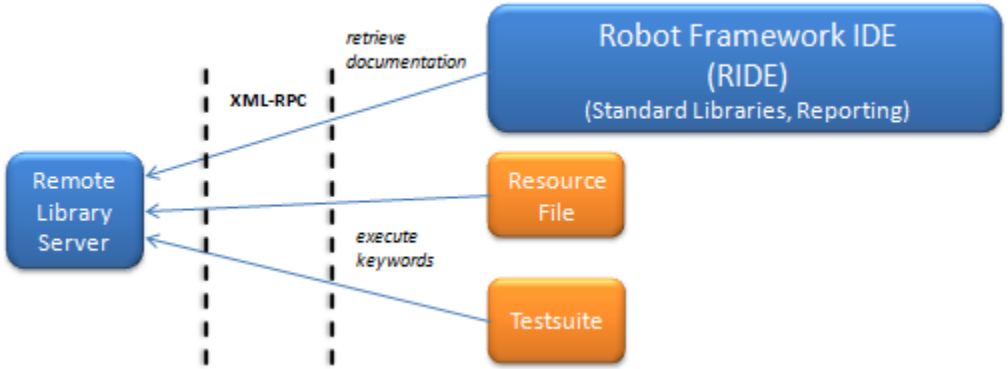
Example – Test Case and Keyword

```
*** Setting ***
Library           OperatingSystem

*** Test Case ***
Windows Directory Not Empty
    [Documentation]    Directory must not be empty.
    [Tags]    windows    directory
    ${FILENUM}=    Number of Files in Directory
    Should Not Be Equal As Numbers    0    ${FILENUM}

*** Keywords ***
Number of Files in Directory
    ${NUM}=    Count Files In Directory    C:\\WINDOWS
    Log    ${NUM}
    [Return]    ${NUM}
```

Accessing Test Libraries remotely using XML-RPC



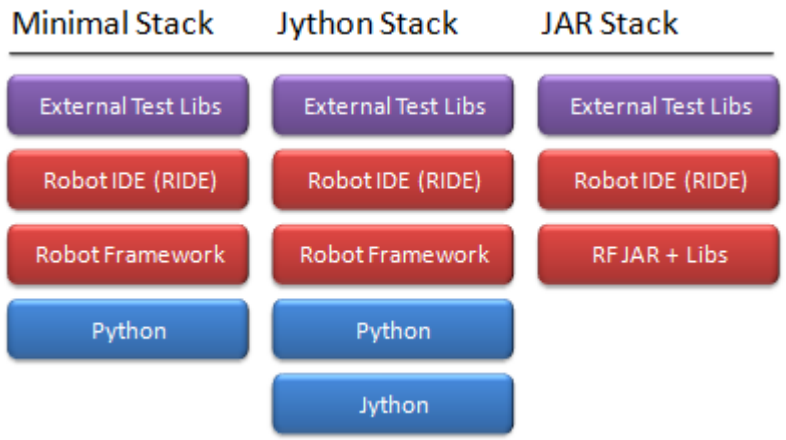

```
Library    org.robot.database.keywords.DatabaseLibrary    <- lokal
Library    Remote    http://localhost:8271                 <- remote
```

http://www.robotframework.org
https://groups.google.com/forum/#!forum/robotframework-users
https://code.google.com/p/robotframework-ride/

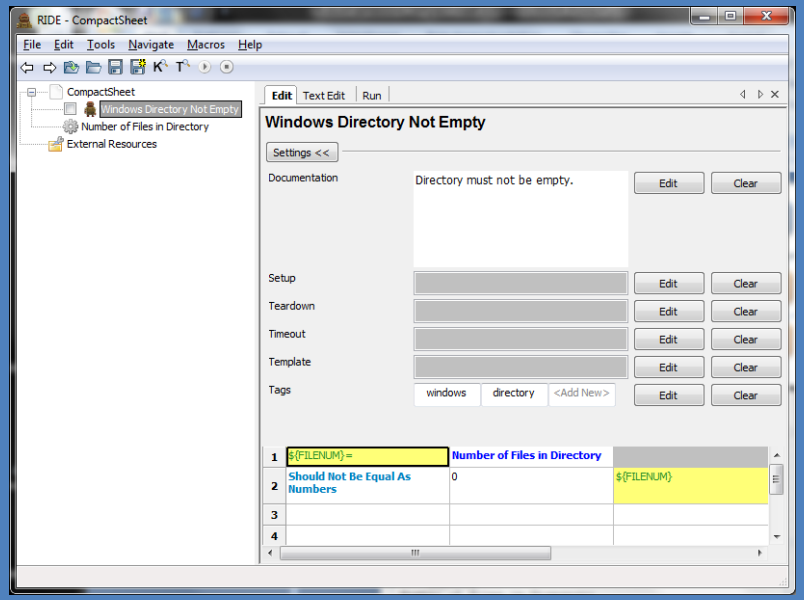
```
pybot|jybot|ipybot [options] data_sources
java -jar robotframework.jar [options] data_sources
```

Robot Framework – *Compact Sheet*

codecentric



RIDE: Graphical User Interface for writing Tests and Keywords. It supports for example auto-completion and syntax highlighting. Allows direct editing of the source files.



Example – Keyword Implementation in Java

```
public class DatabaseLibrary {
    public static final String ROBOT_LIBRARY_SCOPE = "GLOBAL";

    private Connection connection = null;

    public void connectToDatabase(String driverClassName,
                                  String connectString,
                                  String dbUser,
                                  String dbPassword)
        throws SQLException,
               InstantiationException,
               IllegalAccessException,
               ClassNotFoundException {
        Class.forName(driverClassName).newInstance();

        setConnection(DriverManager.getConnection(connectString,
                                                  dbUser,
                                                  dbPassword));

    }

    public void disconnectFromDatabase() throws SQLException {

        getConnection().close();
    }

    public void tableMustExist(String tableName)
        throws SQLException,
               DatabaseLibraryException {
        DatabaseMetaData dbm = getConnection().getMetaData();

        ResultSet rs = dbm.getTables(null, null, tableName, null);

        try {
            if (!rs.next()) {

                throw new DatabaseLibraryException("Table: " +
                                                   tableName +
                                                   " was not found");
            }
        } finally {
            rs.close();
        }
    }
…
```

Robot Framework – *Compact Sheet*

codecentric

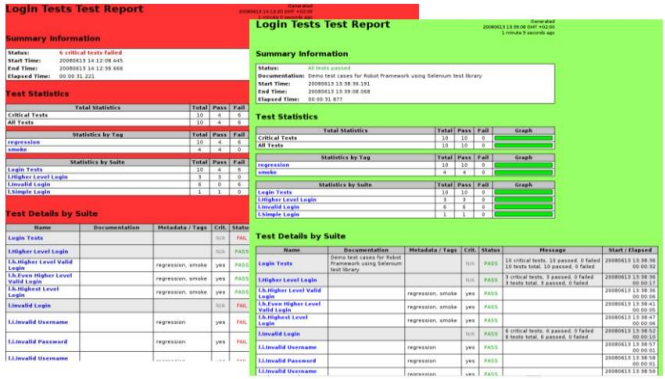
Tools

- **Rebot** - generating logs and reports based on XML outputs and for combining multiple outputs together.
- **Libdoc** - generating keyword documentation for test libraries and resource files.
- **Testdoc** - Generates high level HTML documentation based on Robot Framework test cases.
- **Tidy** - Tool for cleaning up and changing format of Robot Framework test data files.

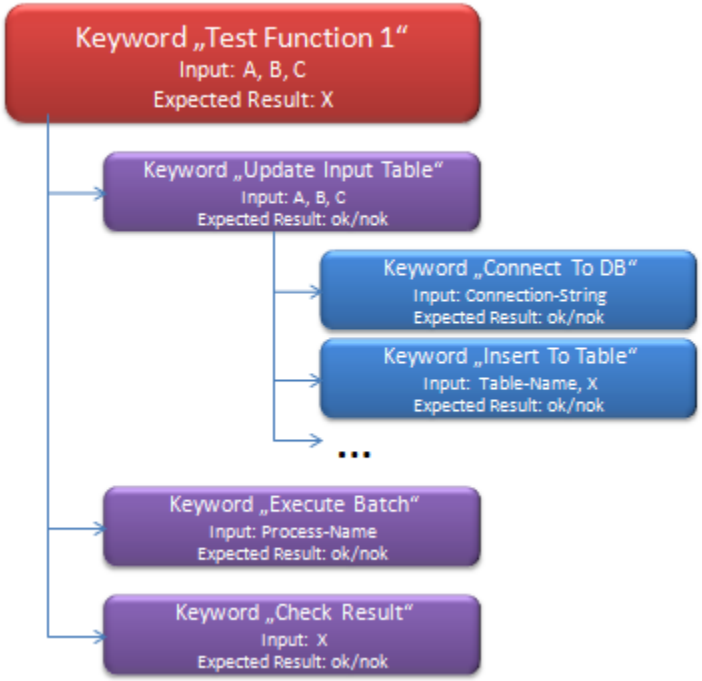
Build Support

- **Jenkins PlugIn** - To collect and publish Robot Framework test results in Jenkins.
- **Maven PlugIn** - For using Robot Framework.
- **Ant Task** - For running Robot Framework tests.

Sample Reports



Combining Keywords to higher-level Keywords



Testing variations using the Generic Testdata Framework AddOn

This AddOn supports testing when a lot of variations should be tested for one functionality. For example in ensurance companies where certain calculations should be tested with a lot of different input values and expected results.

It also supports separation of implementation details from the tests to a greater degree by allowing Tests to be written in Excel.

7	## Calculator Function X Tests		
8	## Scenario	Testcase	Testcase Description
9	calculatorFunctionXTests	Function X Test 1	Description for function x test 1

https://github.com/ThomasJaspers/Generic-Testdata-Framework