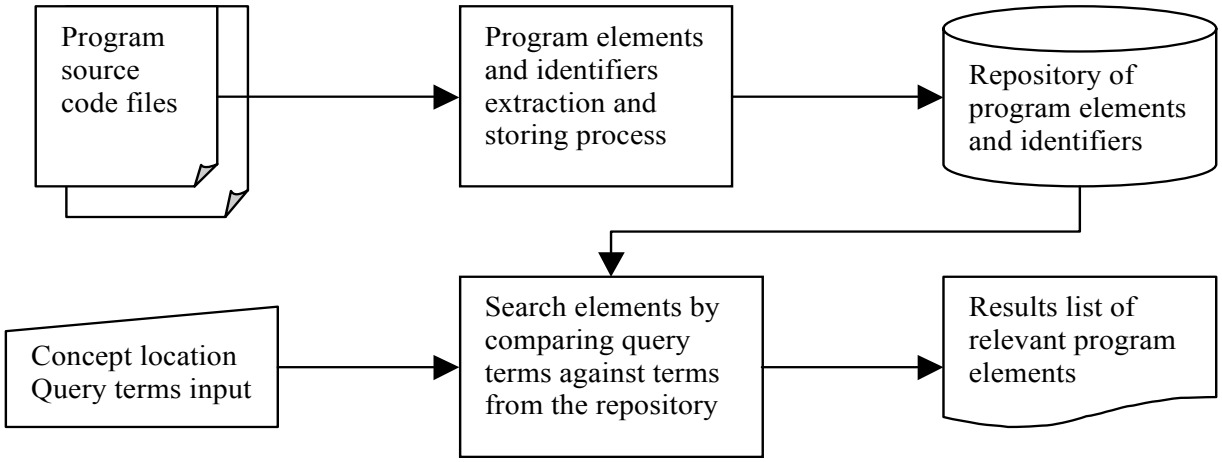
ConCodeSe - Installation and run steps

Abstract
ConCodeSe ranks the source code files of an application using the details (i.e. summary and description) available on a defect document (i.e. bug report) and aims to place the files that may be relevant to fix the defect at the top-10 position of the result list.

Developed based on Information Retrieval, analogous to web searches, first, the source files of the system are processed to extract the required information. Then, given an error report (e.g. taken from an issue tracking system), it suggests a ranked list of files that may have to be changed to fix the error from most to least likely, as illustrated below diagram. This is achieved by utilising heuristics based on contextual information available in certain positions of the bug report summary and description.



For more details please read my publications at http://www.concodese.com/?cat=3

The steps involved are described as follows.

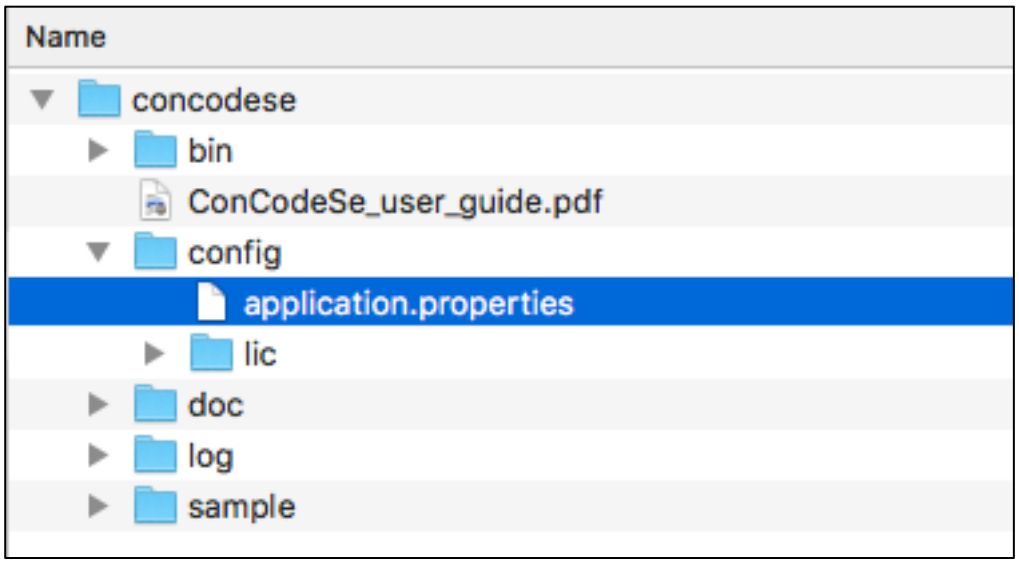
ConCodeSe - Installation and run steps

1. Download, install and configure

1. Ensure that JDK v1.8 or higher is installed and JAVA_HOME on the classpath points to it.

2. If ConCodeSe is already installed continue with step (4), otherwise download the tool from tools category at http://www.concodese.com/

3. Unzip the downloaded file to a location of your choice, e.g. `/opt/app/tools/`. This creates the sub-directory `/concodese`. The image below shows the unzipped directory structure.



4. Open the file *application.properties* in */config* directory in an editor of your choice and specify the following configuration information.
   a) Location of the project source code files.
      e.g. On Unix: `path.to.project=/opt/projects/swt/src_v3.1/`
      On Windows: `path.to.project=C:/projects/swt/src_v3.1/`

   b) Name and version of the project
      e.g, `project.name=SWT`
      `project.version=v3.1`
   Please adhere to the example found in the property file as ConCodeSe utilizes this information.

5. Send an email to info@concodese.com asking for a license file. Please include your name/company

6. You will receive a `license.out` file. Place this file into the `/config/lic` sub-directory under main install directory `/concodese`

Now you can continue with the next sections to create the corpus and then perform searches.

ConCodeSe - Installation and run steps

2. Create a searchable repository

1. From a terminal command line, navigate to the `/bin` directory where ConCodeSe is installed.
   e.g. On Unix: `cd opt/app/tools/concodese/bin`
   On Windows: `cd c:/tools/concodese/bin`

2. Run the following shell command: `./CreateContextualRepository.sh`
   On Windows: `CreateContextualRepository.bat`

   This will run all the steps and create the searchable repository as shown below.

```
ConCodeSe v1.0.0 - Copyright © 2018 Dilshener Consulting. All Rights Reserved.
Started ContextualRepositoryCreator - Sun Oct 07 17:55:11 CEST 2018

[INFO] 2018-10-07 17:55:21,480 [main]  SearchAndRankFilesIDEController.java init - Initializing last DB entry key
[INFO] 2018-10-07 17:55:21,565 [main]  SearchAndRankFilesJIRAController.java init - Initializing headers
[INFO] 2018-10-07 17:55:21,633 [main]  ProjectConfigUpdater.java obtainProjectRunParams - Project run parameters: SWT, v3.1, /Users/tezcan/projects/swt/src_v3.1
[INFO] 2018-10-07 17:55:21,754 [main]  ProjectConfigUpdater.java loadAndPersistProperties - New project properties will be created for SWT v3.1 /Users/tezcan/pro
[INFO] 2018-10-07 17:55:21,908 [main]  ContextualRepositoryCreator.java main - Extracting words from 341
[INFO] 2018-10-07 17:59:16,800 [main]  RepositoryCreationService.java createRepositoryFromSourceFiles - Processing source code words took: 234.890 seconds
[INFO] 2018-10-07 17:59:17,756 [main]  RepositoryCreationService.java createRepositoryFromSourceFiles - Processing source code comments took: 0.955 seconds

Finished ContextualRepositoryCreator - Sun Oct 07 17:59:17 CEST 2018
```

NOTES:

- There may be additional information messages regarding parsing some source code files. These messages can be ignored.

- It is advisable to recreate the searchable corpus regularly, for example each time a new version of the source code is available. In order to do this update the file *application.properties* in */config* directory with the relevant project information, i.e. project's name and version, as descried in Section 1 and repeat above Steps 1 through 2.

PERFORMANCE

Computation time for creating the corpus (i.e. parsing java source code files, comments, stemming and indexing) of an application with 6485 source code files takes about 20min on a machine with 3GHz processor and 16GB RAM.

ConCodeSe - Installation and run steps

3. Perform a search using one of the following use-cases

3.1. Web-based GUI (aimed for IDE search)

Once the repository is created, the user interface to perform a search can be triggered from within an IDE, i.e. Eclipse, as follows.

1. From a terminal (command-line) window, navigate to the `/<install-loc>/concodese/bin` directory and run the following shell command:

   On Unix: `./SearchAndRankFilesServer.sh`

   On Windows: `SearchAndRankFilesServer.bat`

   This will start the ConCodeSe Web Server and ask you to choose the runtime port. Press enter to run it on default port 8080 or specify a different port as shown below.

```
ConCodeSe v1.0.0 - Copyright © 2018 Dilshener Consulting. All Rights Reserved.
Started ConCodeSeJettyServer - Tue Oct 09 10:31:48 CEST 2018
[INFO] 2018-10-09 10:31:49,261 [main]  LicenseChecker.java checkLicenseDateValidity - Valid license -- continue
############################################################
#
# ConCodeSe uses port 8080 for its web-browser based UI.
# If this port is in use, please type an alternative port (e.g. 8082) and enter.
# Otherwise just press enter to continue.
#
############################################################

[INFO] 2018-10-09 10:31:50,478 [main]  Log.java initialized - Logging initialized @1750ms to org.eclipse.jetty.util.log.Slf4jLog
[INFO] 2018-10-09 10:31:50,664 [main]  Server.java doStart - jetty-9.4.z-SNAPSHOT
[INFO] 2018-10-09 10:31:50,854 [main]  AnnotationConfiguration.java scanForAnnotations - Scanning elapsed time=0ms
[INFO] 2018-10-09 10:31:50,861 [main]  StandardDescriptorProcessor.java visitServlet - NO JSP Support for /ConCodeSe, did not fi
[INFO] 2018-10-09 10:31:50,873 [main]  ContextHandler.java log - No Spring WebApplicationInitializer types detected on classpath
[INFO] 2018-10-09 10:31:50,882 [main]  DefaultSessionIdManager.java doStart - DefaultSessionIdManager workerName=node0
[INFO] 2018-10-09 10:31:50,882 [main]  DefaultSessionIdManager.java doStart - No SessionScavenger set, using defaults
[INFO] 2018-10-09 10:31:50,886 [main]  HouseKeeper.java startScavenging - Scavenging every 600000ms
[INFO] 2018-10-09 10:31:50,900 [main]  ContextHandler.java log - Initializing Spring root WebApplicationContext
[INFO] 2018-10-09 10:32:00,589 [main]  SearchAndRankFilesIDEController.java init - Initializing last DB entry key
[INFO] 2018-10-09 10:32:00,788 [main]  SearchAndRankFilesJIRAController.java init - Initializing headers
[INFO] 2018-10-09 10:32:03,373 [main]  ContextHandler.java doStart - Started o.e.j.w.WebAppContext@657c8ad9{/ConCodeSe,jar:file:
ebapp,AVAILABLE}
[INFO] 2018-10-09 10:32:03,406 [main]  AbstractConnector.java doStart - Started ServerConnector@54e11c04{HTTP/1.1,[http/1.1]}{0.
[INFO] 2018-10-09 10:32:03,407 [main]  Server.java doStart - Started @14681ms
```

   Alternatively, you can directly specify a different runtime port with SERVER_PORT argument on the startup script as shown.

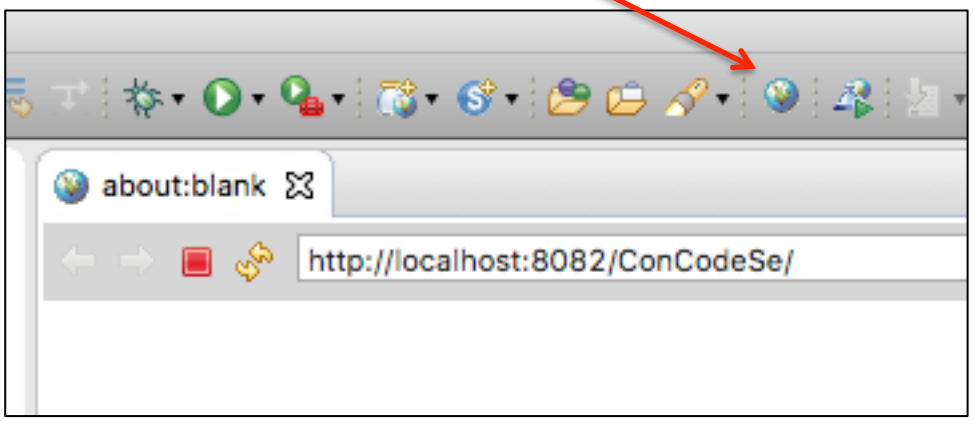
```
^CTezcan-iMac:bin tezcan$ ./SearchAndRankFilesServer.sh SERVER_PORT=8082
ConCodeSe v1.0.0 - Copyright © 2018 Dilshener Consulting. All Rights Reserved.
Started ConCodeSeJettyServer - Tue Oct 09 10:39:36 CEST 2018
[INFO] 2018-10-09 10:39:36,510 [main]  LicenseChecker.java checkLicenseDateValidity - Valid license -- continue
SERVER_PORT specified 8082
[INFO] 2018-10-09 10:39:36,642 [main]  Log.java initialized - Logging initialized @700ms to org.eclipse.jetty.util.l
[INFO] 2018-10-09 10:39:36,832 [main]  Server.java doStart - jetty-9.4.z-SNAPSHOT
[INFO] 2018-10-09 10:39:37,023 [main]  AnnotationConfiguration.java scanForAnnotations - Scanning elapsed time=0ms
[INFO] 2018-10-09 10:39:37,030 [main]  StandardDescriptorProcessor.java visitServlet - NO JSP Support for /ConCodeSe,
[INFO] 2018-10-09 10:39:37,042 [main]  ContextHandler.java log - No Spring WebApplicationInitializer types detected
[INFO] 2018-10-09 10:39:37,051 [main]  DefaultSessionIdManager.java doStart - DefaultSessionIdManager workerName=nod
[INFO] 2018-10-09 10:39:37,051 [main]  DefaultSessionIdManager.java doStart - No SessionScavenger set, using default
[INFO] 2018-10-09 10:39:37,055 [main]  HouseKeeper.java startScavenging - Scavenging every 660000ms
[INFO] 2018-10-09 10:39:37,070 [main]  ContextHandler.java log - Initializing Spring root WebApplicationContext
[INFO] 2018-10-09 10:39:46,625 [main]  SearchAndRankFilesIDEController.java init - Initializing last DB entry key
[INFO] 2018-10-09 10:39:46,727 [main]  SearchAndRankFilesJIRAController.java init - Initializing headers
[INFO] 2018-10-09 10:39:49,547 [main]  ContextHandler.java doStart - Started o.e.j.w.WebAppContext@6295d394{/ConCode
ebapp,AVAILABLE}
[INFO] 2018-10-09 10:39:49,593 [main]  AbstractConnector.java doStart - Started ServerConnector@4a5d1a12{HTTP/1.1,[h
[INFO] 2018-10-09 10:39:49,593 [main]  Server.java doStart - Started @13654ms
```

NOTE: You can terminate the running instance of the web server by pressing key combinations *Cntrl+C*.

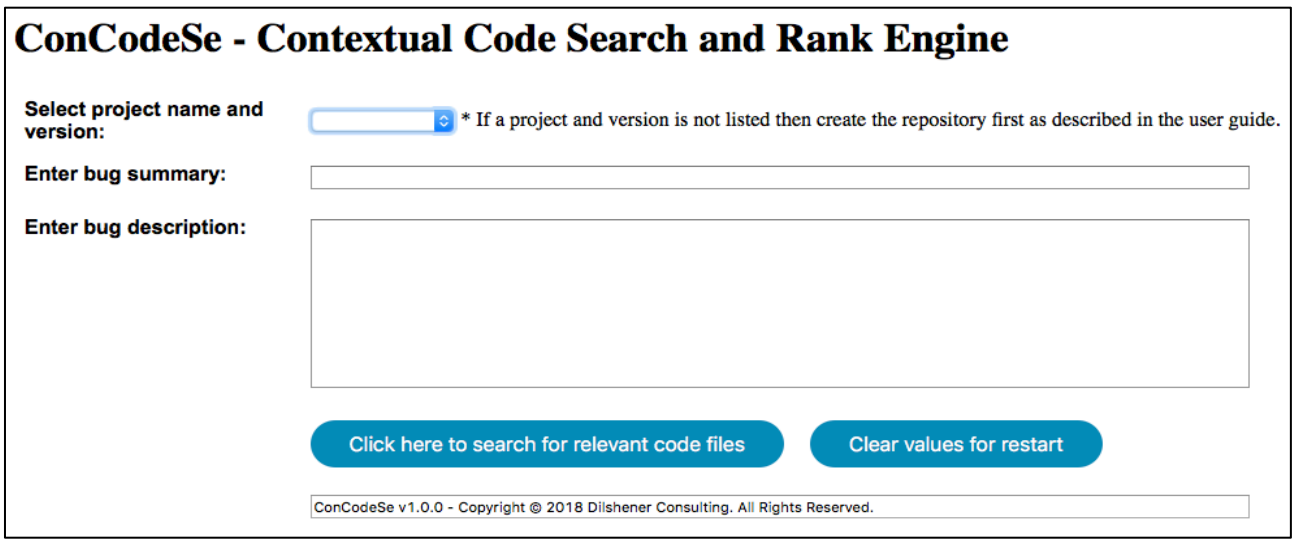
Continue with the next steps found on the following pages.

ConCodeSe - Installation and run steps

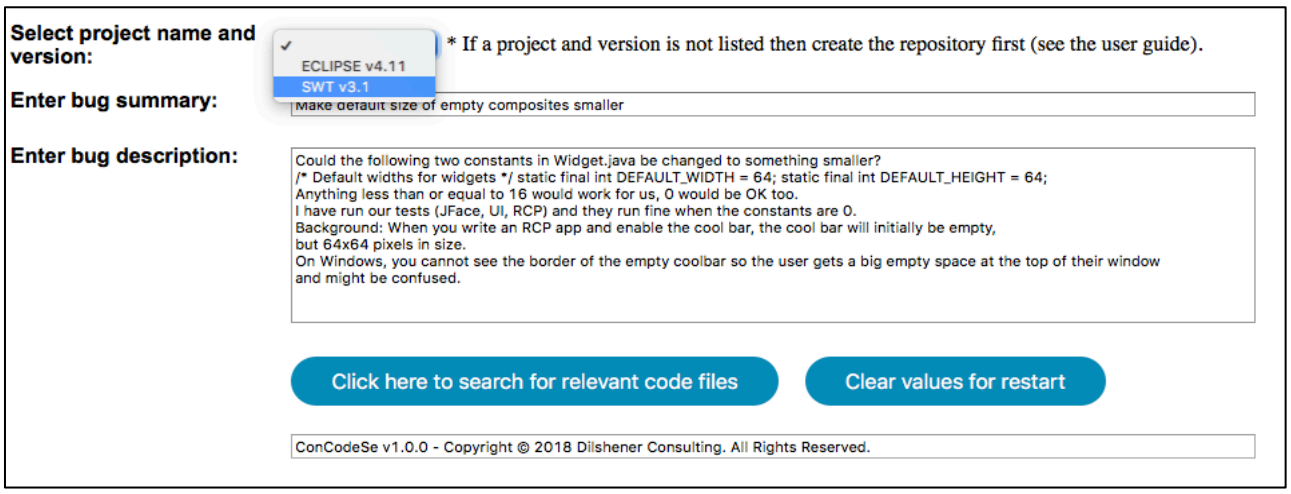
2. In your IDE, open the internal Web Browser, enter the URL to ConCodeSe in the address field as shown below and then press enter to get to the ConCodeSe start page.



NOTE: The port must match the port that was specified previously on startup (Step 1).



3. Select the project name and version from the dropdown menu as shown below. If your project and version is not listed then create it by following the steps in Section 2.

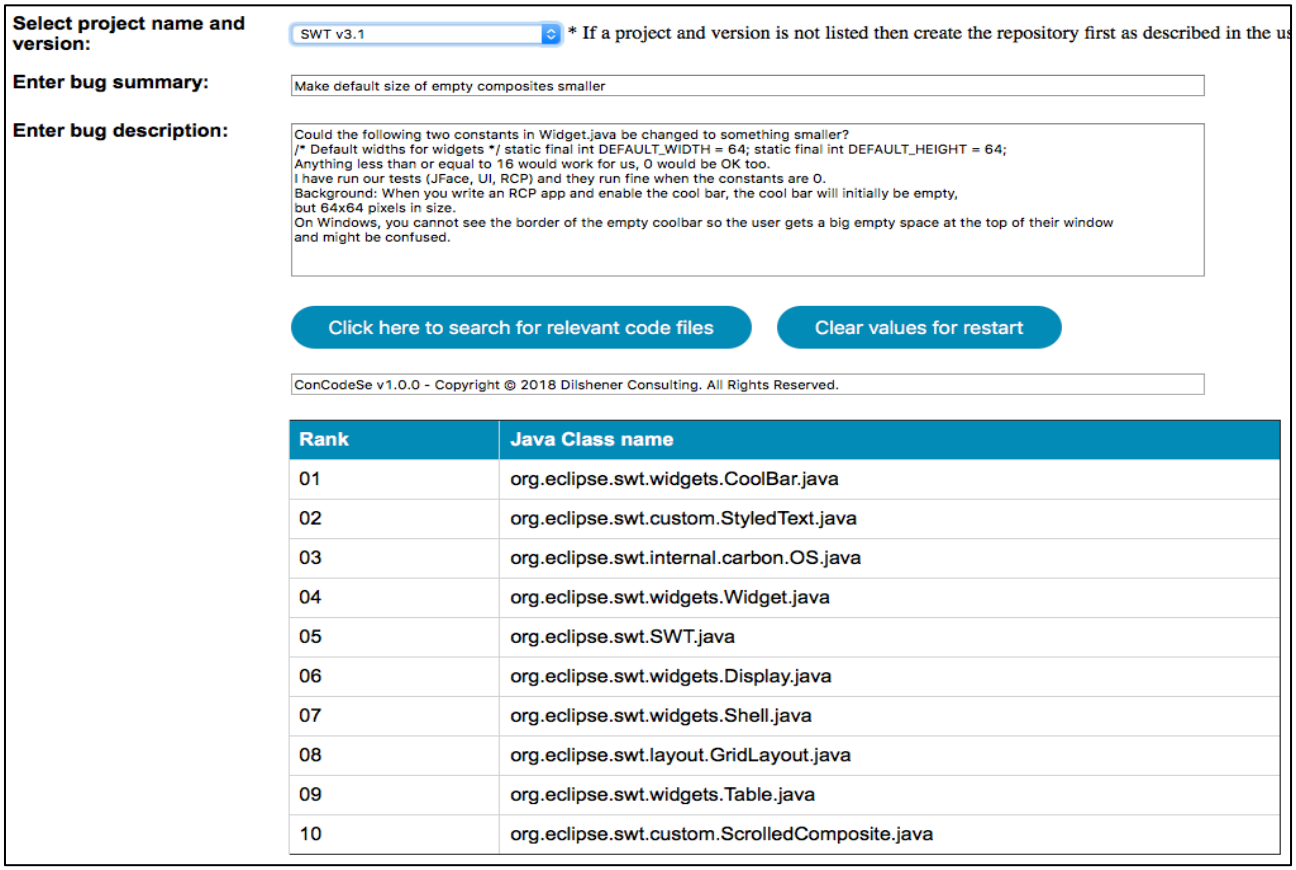


4. Enter the bug report summary and detailed description (e.g stack trace info etc.) as shown above.

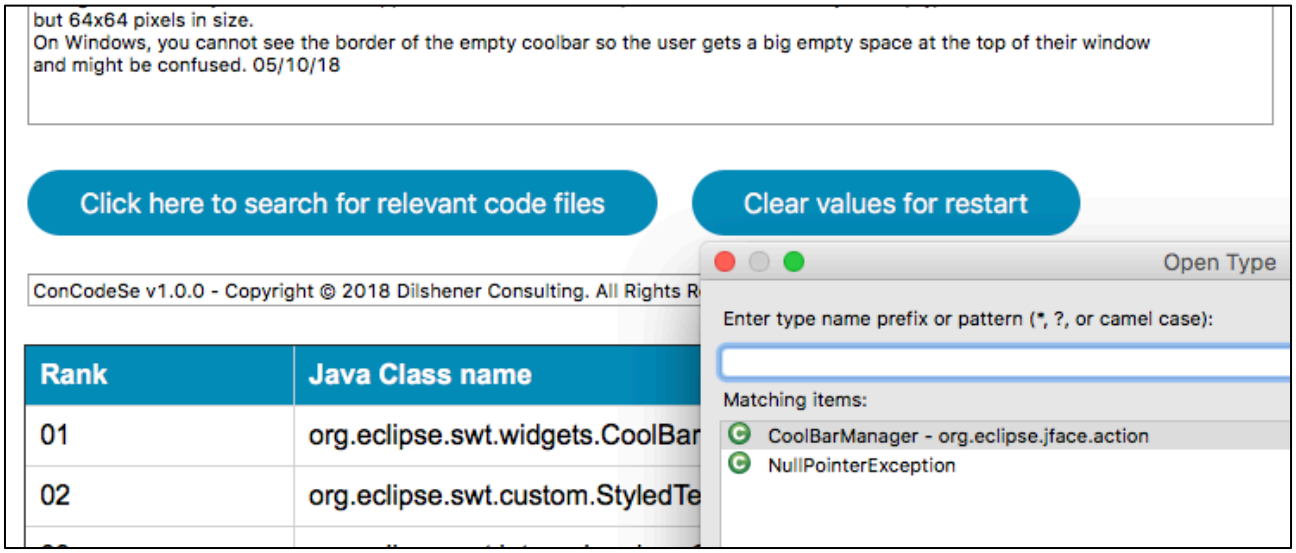
Continue with the next steps found on the following pages.

ConCodeSe - Installation and run steps

5. Click *"search for relevant code files"* button to get a list of classes ranked in the order of relevance to the entered bug summary and description as shown below.



6. Anyone of the candidate classes can be selected from the resulting list and copy/pasted into the Open Type window (ctrl+shift+T) as shown below to select and open its source code file.



ConCodeSe - Installation and run steps

3.2. JIRA integration with the Web-based GUI

ConCodeSe web based UI can also retrieve the summary and description of an existing issue by automatically accessing the JIRA tool using the issue key. Follow the steps to enable this feature.

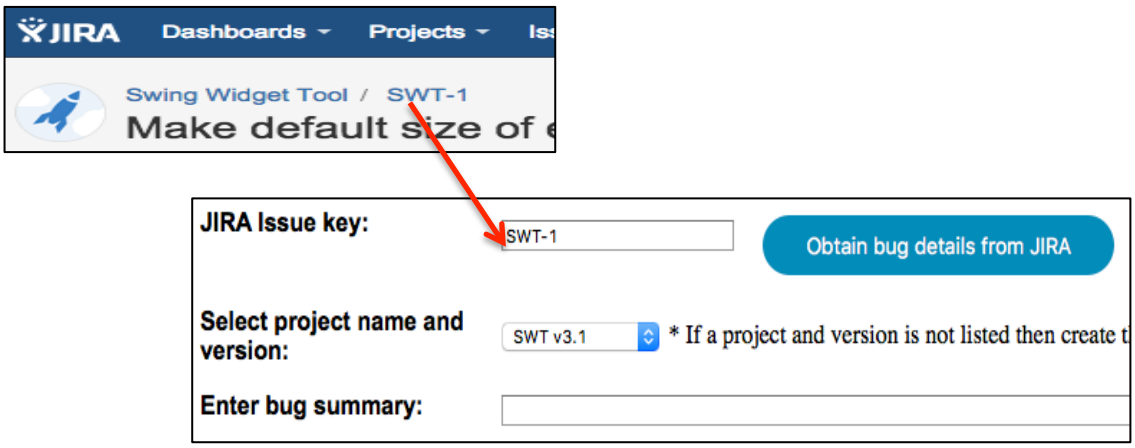
1. Specify the JIRA user, password and URL in the *application.properties* file.
   ```
   jira.user.name=yourJiraUserId
   jira.user.password=YourJiraPassWord
   jira.rest.service.issue.uri=http://localhost:8081/rest/api/2/issue/
   ```
   NOTE: Ensure that project.name defined in Section 2 is the same as the project key defined in JIRA.

2. Stop the ConCodeSe server job if running (enter Cntrl+C from the terminal where its running)

3. Run the following shell command found in `/bin` directory: `./UpdateContextualRepository.sh`
   On Windows: `UpdateContextualRepository.bat`

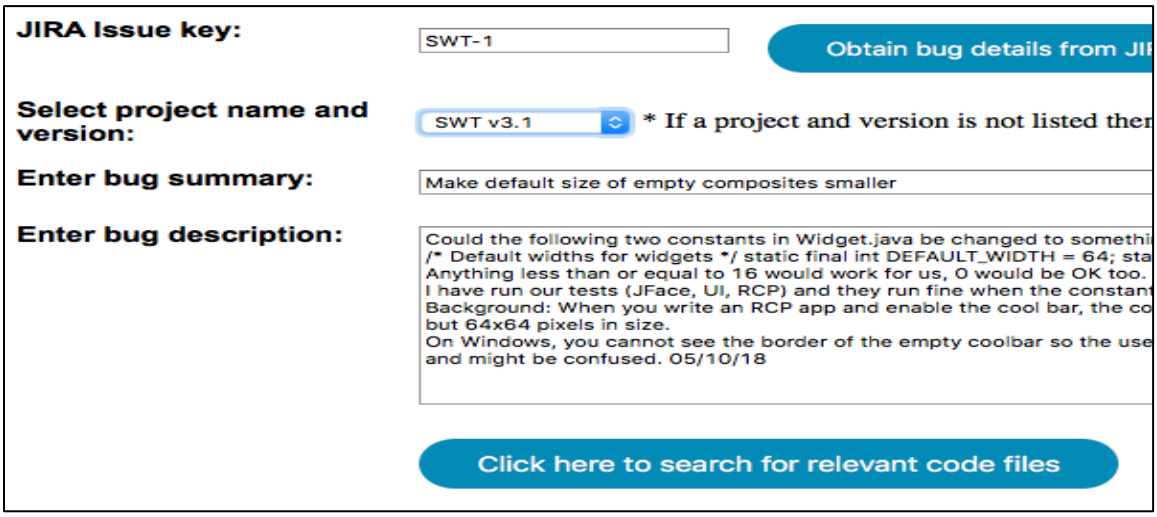
4. Restart the server by performing steps 1 and 2 as described in Section 3.1.

5. Once the search GUI appears, type in the JIRA issue key into the entry field as shown below.



7. Select the project name and version from the dropdown menu as shown below. If your project and version is not listed then create it by following the steps in Section 2.

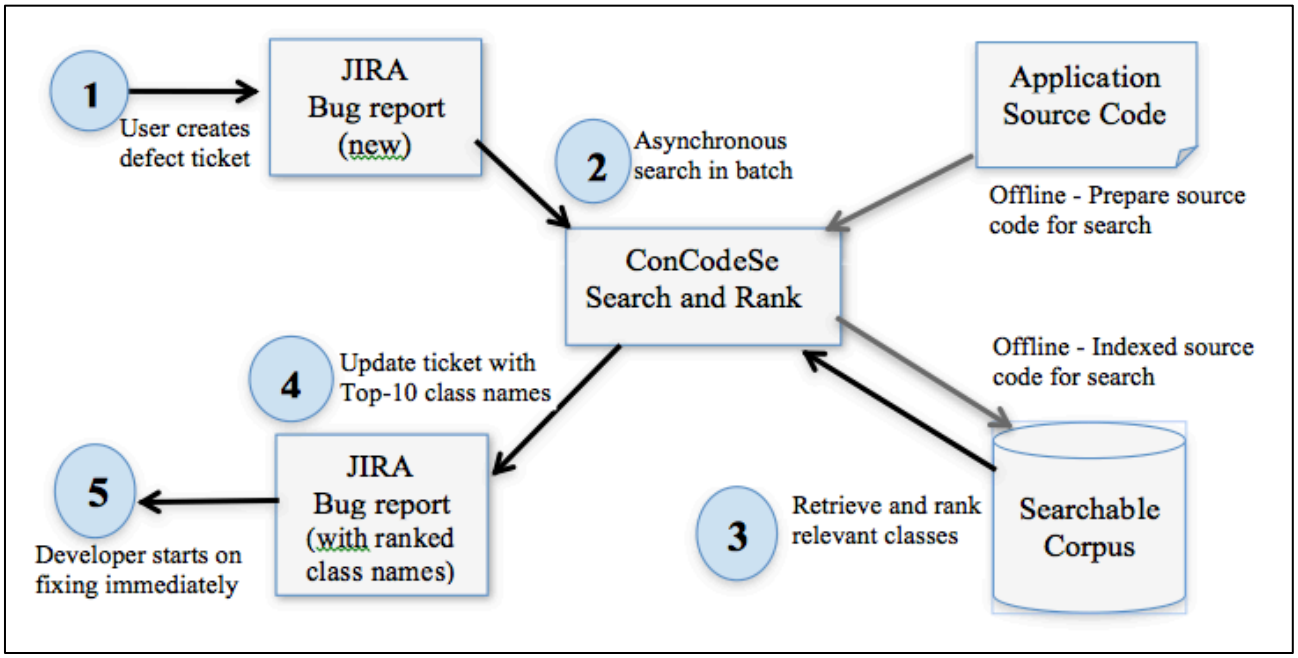
6. Upon pressing "`Obtain bug details from JIRA`" button, the bug summary and description is automatically retrieved from the JIRA as shown below.



7. You can first modify the details, if choose to, and then start a search for relevant source code files.

3.3. Issue tracking tool integration (currently plug-in for JIRA)

ConCodeSe can be integrated into the JIRA issue-tracking tool to perform an automated asynchronous search for candidate classes and then add them into the issue ticket even before the developer starts working on the reported problem. The below diagram illustrates the integration flow.



In order to set up the flow, ConCodeSe needs to be installed either on a standalone server or on the same server where the JIRA application is currently deployed and running. In either case, both applications must be able to reach each other's URL and port. This mean JIRA must be able to reach ConCodeSe web-service URL to start the search (point 2 in the above illustration) and vice versa ConCodeSe must be able to reach JIRA URL to update the JIRA bug report (point 4 in the above illustration).

NOTE: This step may require ports in the server firewall to be opened for ConCodeSe URL (see below example) to be reachable and the install directory of ConCodeSe may need to be declared in your webserver's configuration file. Please consult your system administrator.

e.g. `http://ip.address.of.the.installed.server:portNumber/ConCodeSe/SearchAndRank/`

Furthermore, you need to be familiar with the following two tasks.

1. How to register WebHooks in JIRA
   https://developer.atlassian.com/server/jira/platform/webhooks/
   Section: Registering a webhook via the JIRA administration console

2. Edit your project's workflow to trigger the WebHook upon transition of states, i.e. created, updated.
   https://confluence.atlassian.com/adminjiraserver/working-with-workflows-938847362.html
   Section: Editing a project's workflow
   Note: This step may only be necessary if your project's workflow deviates substantially from the standard built in workflows of JIRA.

Continue with the next steps found on the following pages.

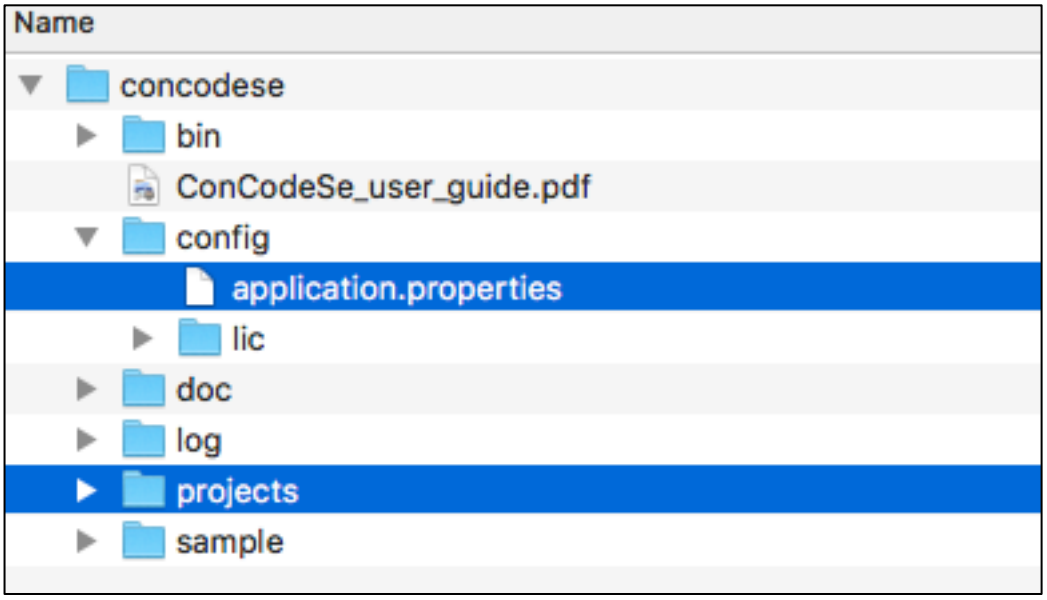
ConCodeSe - Installation and run steps

1. Perform steps described in Section 1 and unzip the downloaded ConCodeSe file to a location of your choice on the server, e.g. `/opt/app/tools/`.

2. Create a dedicated user in JIRA for ConCodeSe to update issue tickets.
   e.g. `jira.user.name=ConCodeSe`
   `jira.user.password=SearchAndRank10`

4. Specify the JIRA user, password and URL in the *application.properties* file.
   e.g. `jira.user.name=ConCodeSe`
   `jira.user.password=SearchAndRank10`
   `jira.rest.service.issue.uri=http://localhost:8081/rest/api/2/issue/`
   Note: Ensure that `project.name` property defined in Section 2 is the same as the project name defined in JIRA for which the ConCodeSe is being configured. Otherwise update it in the file.

5. Choose to perform one of the following options:
   a. If the source code of the application is not indexed
      i. Perform steps described in Section 2 to create the searchable corpus.
   b. If the source code is already indexed on a different machine,
      i. Copy the `/projects` sub-directory from there to this server.



6. Run the shell command found in `/bin` directory: `./UpdateContextualRepository.sh` to update the project's ConCodeSe runtime configurations.

7. Start the ConCodeSe server job by running the shell command: `./concodese_service.sh start`
   This will start the ConCodeSe Web server on port 8080 as headless background job. The process ID and the log info will be written to their dedicated files in the `/log` sub-directory as shown below.

7. You can specify a different port using the SERVER_PORT argument on the startup script:
   On Unix: `./concodese_service.sh start SERVER_PORT=9200`

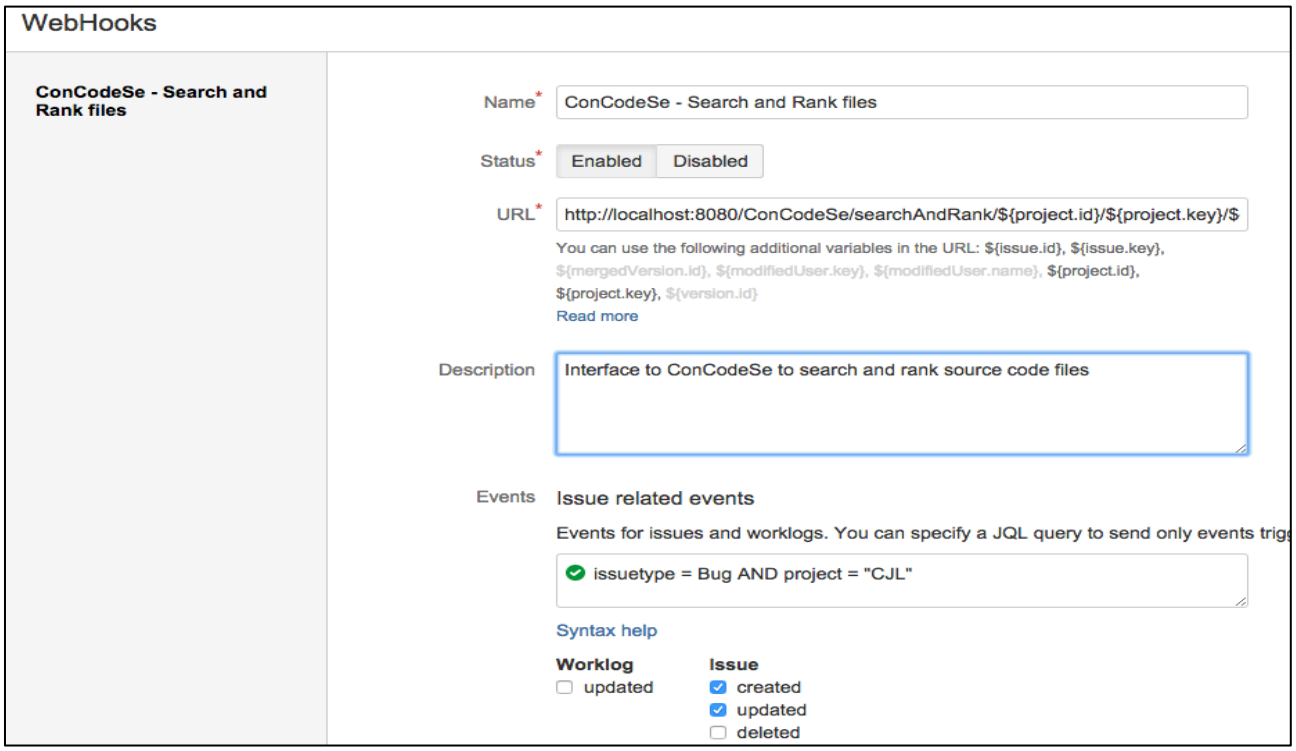
8. You can stop the service by running: `./concodese_service.sh stop`
   Other usages are: `./concodese_service.sh { start | stop | restart | status }`

Continue with the next steps found on the following pages.

9. Login to JIRA tool as an administrator and create a web-hook to ConCodeSe REST service as:

URL: *http://localhost:8080/ConCodeSe/searchAndRank/${project.id}/${project.key}/${issue.id}/${issue.key}*
Event: *issuetype = Bug AND project = <your_project_name>*
Issue: *created AND updated*

Note: The host and port that ConCodeSe is running on must be reachable from the JIRA when creating and the order of the arguments on the URL is important as shown in the illustrations below.



Note: If ConCodeSe is being configured for multiple JIRA projects, the event JQL can use ***in*** syntax like:
issuetype = Bug AND project in (project_name_1, project_name_2)

Continue with the next steps found on the following pages.

10. Create an issue in JIRA by entering a summary and detailed description as shown below.

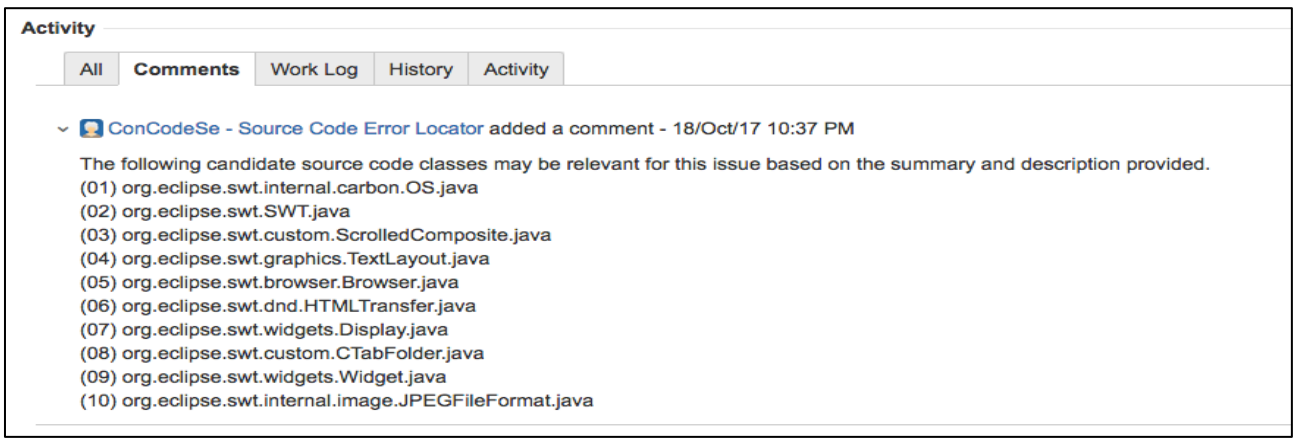
6. Upon pressing the create button in JIRA, the ConCodeSe rest service is called in the back ground with the project/issue - key/id values. The rest service asynchronously calls back the JIRA with the issue id to obtain the bug details as shown in the log file.

```
INFO  - Server                          - Started @6823ms
INFO  - archAndRankFilesJIRAController - projectId: 10000 / projectKey: CJL / issueId: 10204 / issueKey: CJL-8
INFO  - archAndRankFilesJIRAController - JIRA issue info uri: http://localhost:8081/rest/api/2/issue/10204
INFO  - ndRankFilesFrontendServiceImpl - bugReport: BugReport[projectId=10000,projectKey=CJL,issueId=10204,issueKey=CJL-8,summary=GDI+ drawStri
ng is not consistent with GDI,description=If you have a structure which is rendered to a GC, you need to be able to predict things like how lar
ge a String is going to be painted.
But, if a previous visitor has triggered GDI+, drawString's behavior is altered.
The chars get expanded and compressed in funny ways, and generally look bad. ,versions=[v1]]
INFO  - ChangeRequestGUIExtractorImpl  - Extracting Change Request summary and description words.
INFO  - ChangeRequestGUIExtractorImpl  - Processed Change Requests
```

7. Subsequently a search is performed in ConCodeSe using the summary and description obtained from the bug details. Finally, the rest service calls back the JIRA rest api to update the comment field of the issue with the candidate classes as shown below in the log file.

```
INFO  - archAndRankFilesGUIServiceImpl - Searched and ranked files size: 27
INFO  - archAndRankFilesJIRAController - uriUpdate: http://localhost:8081/rest/api/2/issue/10204/comment
INFO  - archAndRankFilesJIRAController - Updated issueId: 10204 - Response from JIRA: 201:Created
INFO  - archAndRankFilesJIRAController - DONE
```

8. Refresh the JIRA issue to see the comment field updated with the list of classes as shown below.



ConCodeSe - Installation and run steps

4. Keeping the searchable repository up-to-date

Each time a new version of the source code is checked out into the local repository, i.e. local workspace, it is advisable to rebuild the searchable corpus as follows.

1. From a terminal command line, navigate to the directory where ConCodeSe is installed.
   e.g. On Unix: `cd opt/app/tools/concodese`
   On Windows: `cd c:/tools/concodese`

2. Update the project details, e.g. project name and version, in the *application.properties* file found in */config* directory (see Section 2. Create searchable corpus).

3. Repeat steps described in Section 2. Create searchable corpus.

4. Perform one of the search options as described in Section 3.