

Hourglass Smart Test Job Runner

Hourglass Software LLC



Contact: support@hourglass-software.com

About

Hourglass Smart Test Job Runner is a tool that enhances and optimizes your CI/CD Pipelines. It uses subcomponents and services for risk analysis and quality assessment tasks, also using Machine Learning, and determinations where the resulting data finds areas of risk in your software and executes CI/CD Pipeline jobs accordingly. This targets CI/CD Pipeline jobs that are test automation. The results from the risk assessment will prioritize which test automation jobs to execute in priority order and only those that have been assessed to have test coverage for software areas that are “risky” and have a higher potential to yield defects. Thus, it is an efficient means of managing your CI/CD pipeline test automation jobs.

Systems Integration

The tool runs on Windows from command line. It integrates (requires) the following systems:

1. **Atlassian Jira** (for Machine Learning processing Bug history within areas of software under test)
2. **BitBucket** or **Github** (for using statistical data analysis on commits and changes to the repository to yield another set of risk assessment)
3. **Jenkins** or **TeamCity** (or any other CI/CD system that can trigger job execution using Web Requests via REST)
4. Runs on Windows (Command line)

System Requirements

The tool requires to following on your Windows host

1. Java 8 / JDK 8
2. .NET Core 3.1
3. NodeJS

How it works

Once executed, the application performs the tasks every 12 hours (application must be running). It runs/uses a Java application using Machine Learning (Hourglass Bug Predictor) to predict bug counts for specified areas in your software under test and results with bug prediction counts (Jira is required). It then runs “Code Risk Analyzer” which is a NodeJS application that uses either Github or BitBucket to query statistical data on code commits and code changes, and results with which source code files have the highest “risk” of yielding defects. Then, based on the results of these both, the application will execute test automation jobs in CI/CD (ie: Jenkins or TeamCity) that are associated with the software areas that have the highest probability for defects to occur. The results of “Hourglass Bug Predictor” and “Code Risk Analyzer” are normalized and combined.

Thus, as an example, if there are 100 CI/CD jobs that are in place to run, every 12 hours this tool will automatically only run the ones (10 jobs, for example) that have the highest probability of finding defects. This is a “smart” and efficient means of triggering test automation jobs in pipelines.

(see below for how “Hourglass Bug Predictor” and “Code Risk Analyzer” work)

Setup

- Simply run the MSI installer on Windows
- DO NOT change the installed files and folders installed or any related directories and directory structure and location
- DO NOT change any file or directory names
- DO NOT change the environment variable set for “HOURGLASS_BUG_PREDICTOR”

Background

- The application requires following data:
 - An integer ID for each of the CI/CD test automation jobs
 - A mapping between source files or folders to the job ID
 - A mapping between Jira data to job ID
 - A mapping for job ID to the test automation trigger REST URL into your CI/CD jobs triggered remotely

Required Configuration files

- \Hourglass Smart Test Job Runner\Services\HourglassBugPredictor\data the following 2 files are required (see “Hourglass Bug Predictor” section for details):
 - bug-predictor-schema.xml (sample provided – replace this)
 - predictParamData.csv (sample provided – replace this)
- \Hourglass Smart Test Job Runner\mappings\files, create the following file:
 - job_to_file_mapping.csv (sample provided, replace this)
 - 2 values separated by comma on each line
 - First column is job ID number

- Second column is source file name of folder name in source file path that is associated with the job ID (see sample)
 - Can have as many entries as you like
 - Job IDs can be repeated for more than 1 file (entry)
- \Hourglass Smart Test Job Runner\mappings\jobs, create the following file:
 - Job_to_webhook_mapping.csv (sample provided, replace this)
 - Each line has 2 values separated by comma
 - First value is job ID
 - Second value is job triggering URL used in Web Request to trigger (see sample)
 - Can be a parameterized URL
 - Only 1 entry for job ID to URL allowed

Running the Application

Once you have installed the files and folders and have setup the configurations required, simply run the tool from command line with:

Go to folder with executable:

\Hourglass Smart Test Job Runner\Controller\HourglassSmartTestJobRunner

Execute:

```
.\HourglassSmartTestJobRunner.exe <jira url> <jira user> <jira auth token or password> <github/bitbucket> <number of jobs to trigger. Ie: top X> <CI/CD user- agent user> <CI/CD user-agent token or password>
```

If using github, append command line parameters:

```
<github repo> <github user> <github token>
```

If using bitbucket:

```
<bitbucket url> <bitbucket project> <bitbucket repo> <bitbucket token>
```

All command line parameters must be 100% valid.

Example for github:

```
.\HourglassSmartTestJobRunner.exe "https://hourglass-software.atlassian.net" "faisal@hourglass-software.com" "xx112233" "github" 2 "fizzq" "ababab" "Fizz-Q/code-risk-analyzer" "Fizz-Q" "xyxyxyxy"
```

Once executed, let the application continuously run. It will trigger jobs every 12 hours.

Hourglass Bug Predictor

This is the first of the two services that assess software quality and risk. (uses Machine Learning)

[.\Hourglass Bug Predictor.pdf](#)

Hourglass Code Risk Analyzer

Second service to perform risk and quality assessment

[.\Hourglass Code Risk Analyzer.pdf](#)