Rational Build Forge Incubator for Platform Metrics Tool

Static Analysis Tool

Kristofer Duer

July 23, 2010

Introduction

The Rational Build Forge Incubator for Platform Metrics Tool, henceforth referred to as RBFIPMT, is designed to be a low impact static analysis tool of a Build Forge environment. The information it gathers is specific to Build Forge artifacts and their statistical information within a Build Forge database. The statistical information this tool is concerned with centers around the following categories:

- Total
- Average
- Maximum value
- Minimum value
- Standard Deviation
- Total count of parent artifacts processed

One of the primary goals of the tool is to diagnose and report on usage of Build Forge without including any customized information – such as project names. Additionally the tool is able to be or throttled down for less of an impact on a production environment. Finally the tool reports what is being gathered currently, as well as whether the API was used or the file system if this is a restart.

Example:

Grouped By Group Owner :
Total: 16.0
Average: 8.0
Maximum Value: 8.0
Minimum Value: 8.0
Standard Deviation: 0.0
Number of Parent Artifacts Processed: 2.0

In this example an access group statistic is displayed. In this statistic no group names, user memberships, or any subgroups associated with the access group are listed – only numbers.

Requirements:
The tool requires Build Forge 7.1.1.1 or later to operate properly, as well as Java 1.5 or later.

Using the Rational Build Forge Incubator for Platform Metrics Tool

This section covers how to install and use the tool.

How it works

The RBFIPMT connects to the Build Forge instance specified in the connection parameters and gathers the statistical information based on the profile mode requested. The flow the tool follows is:

1) Build the list of artifact uuids to profile
    a. This acts as a point in time snapshot of the current state of Build Forge. Artifacts can be deleted, however new artifacts added will not be contained within this current profile run.
2) Gather statistics based on the list of gathered uuids for the requested profile mode
    a. Each artifact gathered will produce an xml file within a folder inside the root path. This is used for persistence purposes and is deleted when the RBFIPMT completes
3) Build reports based on the gathered statistics
4) Zip the reports into a file called BFReports_<CURRENT DATE>.zip
    a. For example BFReports_6_8_2010.zip
5) Clean up the root directory of the profile run except for the zip file and profiler.log

The RBFIPMT will write files to disk as it moves through each artifact. Once the profile is gathered all that will be left is a zip file and the log file – profile.log. If the Profiler is run a second time in the same day the next zip file name will be BFReports_<CURRENT DATE>_<N>.zip where N is the current number of report zip files in the directory – for example BFReports_6_8_2010_2.zip.

The RBFIPMT stores the current status of the gathered Build Forge artifacts in folders within the root path. If –all is used the folders contained within the root path will be similar to:

```
07/08/2010  07:17 PM    <DIR>          .
07/08/2010  07:17 PM    <DIR>          ..
07/08/2010  07:16 PM    <DIR>          AccessGroupDetails
07/08/2010  07:16 PM    <DIR>          AdaptorDetails
07/08/2010  07:16 PM    <DIR>          AdaptorLinkDetails
07/08/2010  07:17 PM    <DIR>          BuildBomDetails
07/08/2010  07:17 PM    <DIR>          BuildBomManifestDetails
07/08/2010  07:16 PM    <DIR>          BuildDetails
07/08/2010  07:16 PM    <DIR>          BuildEnvironmentDetails
07/08/2010  07:16 PM    <DIR>          BuildEnvironmentResultDetails
07/08/2010  07:17 PM    <DIR>          BuildLogDetails
07/08/2010  07:16 PM    <DIR>          BuildResultDetails
07/08/2010  07:16 PM    <DIR>          ClassDetails
07/08/2010  07:16 PM    <DIR>          CollectorDetails
07/08/2010  07:16 PM    <DIR>          EnvironmentDetails
07/08/2010  07:16 PM    <DIR>          FilterDetails
07/08/2010  07:16 PM    <DIR>          Profiler
07/08/2010  07:16 PM    <DIR>          ProjectDetails
07/08/2010  07:17 PM    <DIR>          Reports
07/08/2010  07:16 PM    <DIR>          ScheduleDetails
07/08/2010  07:16 PM    <DIR>          SelectorDetails
07/08/2010  07:16 PM    <DIR>          ServerDetails
07/08/2010  07:16 PM    <DIR>          StepDetails
```

```
07/08/2010  07:16 PM    <DIR>          SysConfigDetails
07/08/2010  07:16 PM    <DIR>          TemplateDetails
07/08/2010  07:16 PM    <DIR>          UserDetails
```

Inside each of these directories is an XML representation of the Proflier object. These files are used to persist and grant the ability to stop a running profile and restart it at a later time. If the files get corrupted, or deleted between restarts of the RBFIPMT the tool will simply rebuild the file and start from the beginning.

When the tool is complete all that will be left are zip files and a single profiler.log file containing the artifacts uuid's which were processed during the profile gathering.

Installation

The RBFIPMT is intended to be used form a client machine, and the Build Forge Java API to collect information. It is not intended to be run directly from the management console, however this can be done if desired.

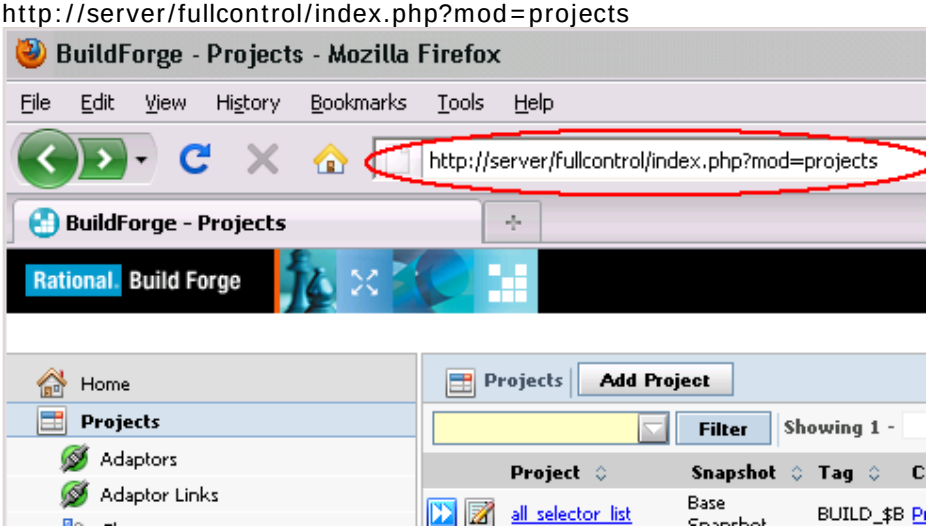
The rbf-services-client-java.jar from the currently installed Build Forge instance to be profiled is required to be in the same directory where the Profiler.jar resides for proper functioning.

Retrieving the client jar

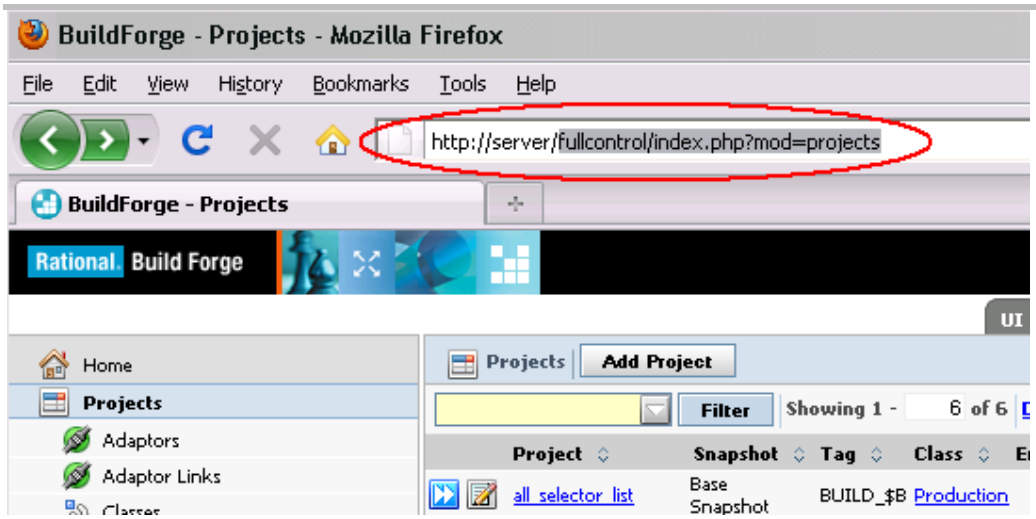
The rbf-services-client-java.jar can be retrieved in one of two ways:

1) From a web browser

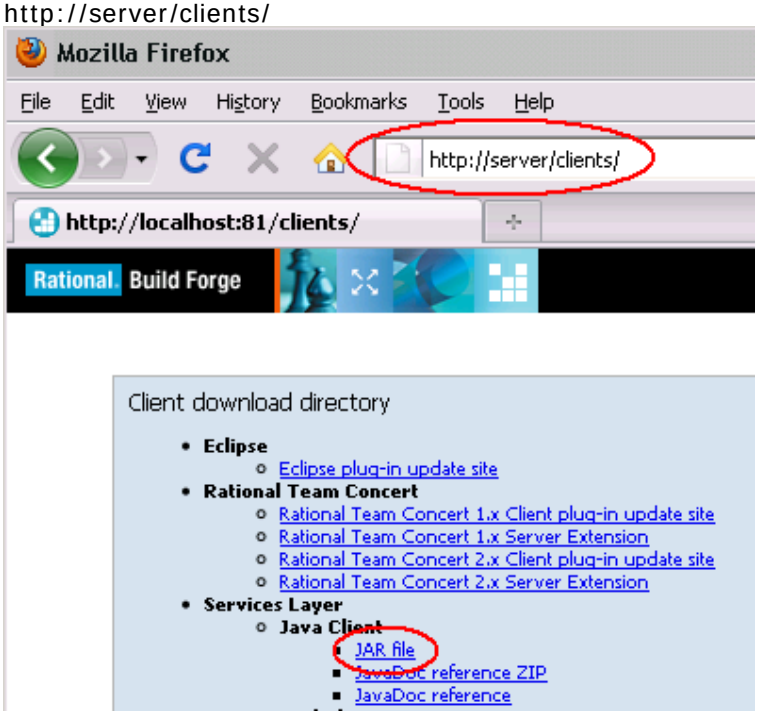
To retrieve rbf-services-client-java.jar from a web broswer first log in to the Build Forge management console. This will produce a URL at the top which appears similar to:
http://server/fullcontrol/index.php?mod=projects



Replace everything after fullcontrol/ including full control with the keyword clients. For example:
http://server/clients

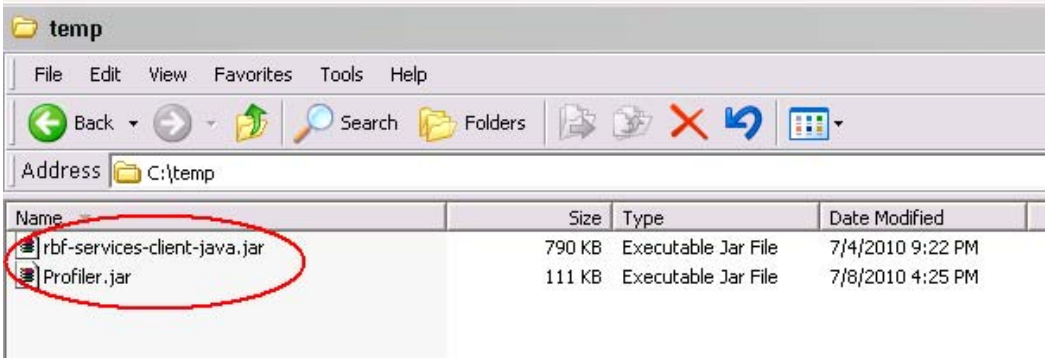


This will bring up a page which contains all of the API clients and documentation. The new URL appears similar to :
http://server/clients/



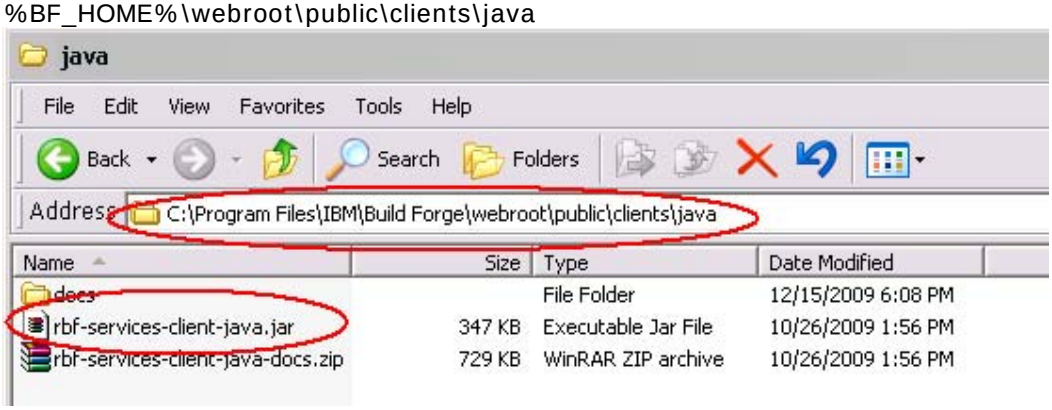
Click on the link for the Jar File under Java. This link looks similar to:
http://server/clients/java/rbf-services-client-java.jar

Place this in the same location as the Profiler.jar file

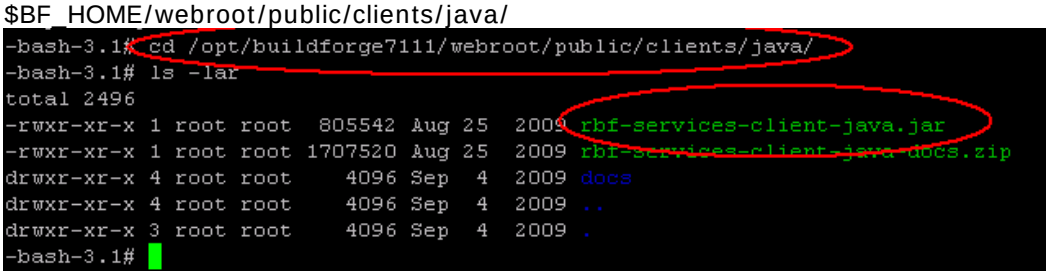


2) From the management console file system

This method requires access to the file system of the Build Forge management console. If you do not have access to the Build Forge management console use option 1.
On Windows the rbf-services-client-java.jar is located in:
%BF_HOME%\webroot\public\clients\java



On UNIX the rbf-services-client-java.jar is located in:
$BF_HOME/webroot/public/clients/java/



In either case copy the file locally to the machine which will be running the RBFIPMT and place the rbf-services-client-java.jar and Profiler.jar file in the same directory.

For example:
C:\bf_src\perf\profiler\src>dir
Volume in drive C is C_Drive
Volume Serial Number is 126C-EAB8

Directory of C:\temp

```
07/08/2010  02:47 PM    <DIR>          .
07/08/2010  02:47 PM    <DIR>          ..
07/08/2010  02:47 PM           112,810 Profiler.jar
07/04/2010  09:22 PM           808,571 rbf-services-client-java.jar
               5 File(s)      1,038,123 bytes
               5 Dir(s)  20,002,873,344 bytes free
```

Usage

The RBFIPMT options are broken down into three categories:
1) Connection parameters
2) Profile mode
3) Extra options

java -jar Profiler.jar {Connection Parameters} {Profile Mode} [Extra Options]

The connection parameters and profile mode must be entered for the RBFIPMT to run. The extra options are optional.

Connection parameters:

The connection parameters are used to guide the RBFIPMT to the currently running Build Forge instance. The connection options are passed to the internal Build Forge Java API connection object. There are two required entries, and three optional ones:

Required connection parameters:

1) Username
    a. The Build Forge username for use with the API
2) Password
    a. The Build Forge password for the username specified in #1

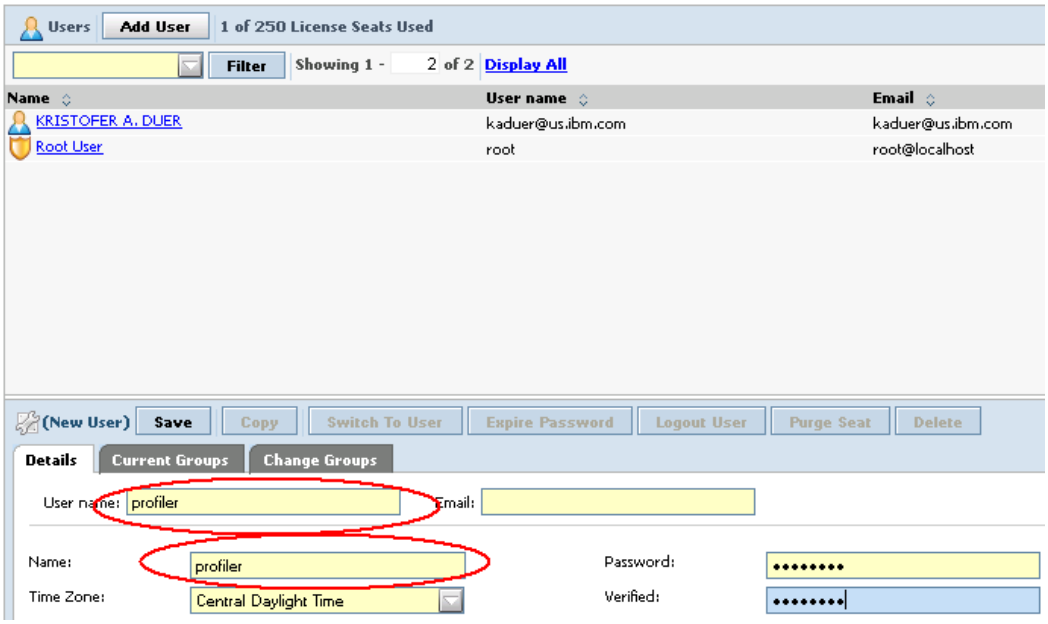
Optional connection parameters:

1) BF Console host
    a. Specifies the host to connect to when running the RBFIPMT. This will be the machine where Build Forge is installed. The default is localhost.
2) BF API port number
    a. This is the services layer port currently being used by the installation. The default is 3966.
3) BF API user domain
    a. This is the logical domain name to use if the API user is an LDAP user

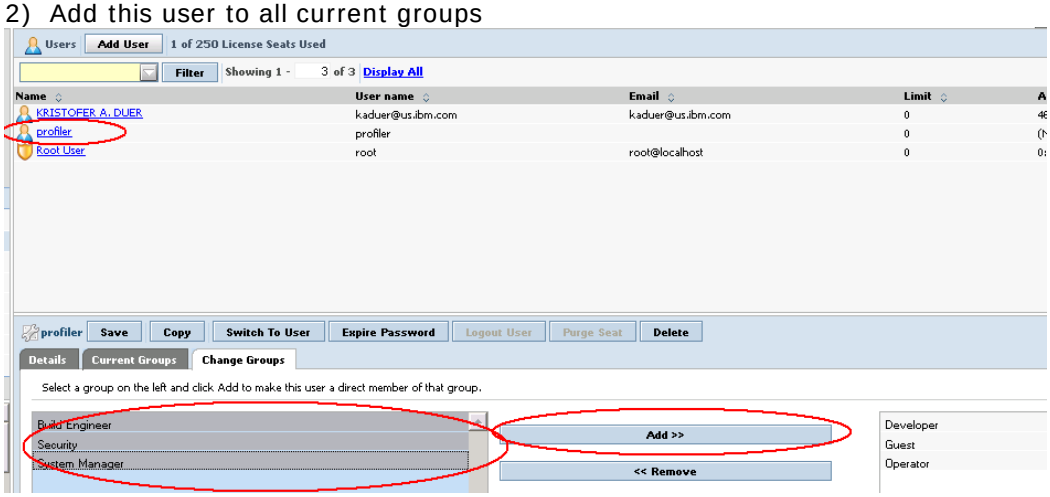
Note: The RBFIPMT will only be able to see and profile those Build Forge artifacts the user has access to. Avoid using the root user for this tool as a best practice, and instead create a Build Forge API user and add group membership to all current groups, or those groups which control the artifacts you wish to profile. If you do not add all the groups to the user profiling the environment the information will be incomplete.

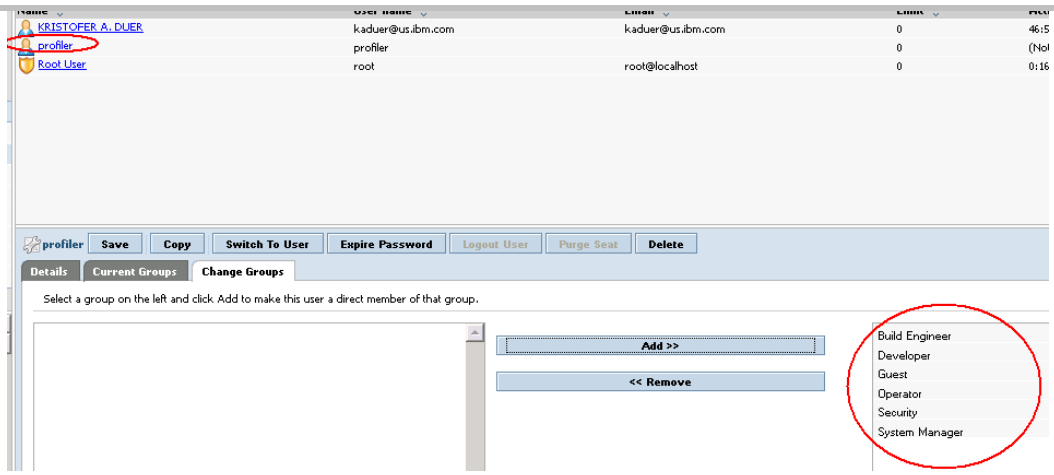
Adding a RBFIPMT API user

1) Create a new Build Forge user. In this example the user profiler with the password profiler is created. In a production environment it is recommended to use a more secure password. The username is also able to be changed if profiler is not the preferred name.



2) Add this user to all current groups





A profiler user is now ready for use within this Build Forge environment and the RBFIPMT.

Profile Mode:

The profile mode controls what Build Forge artifacts are profiled. This can be useful in targeting just configuration data, just build data or build log sizing data.

There are four choices for profile mode:

3) Configuration Analysis
4) Build Analysis
5) Build Logs Analysis
6) All

Configuration Analysis

The configuration analysis mode will profile the configuration artifacts listed below. Only configuration artifact static analysis is done in this section. A complete list of all analysis done for configuration analysis is available in Appendix A.

1) Access Groups
2) Adaptors
3) Adaptor Links
4) Classes
5) Collectors
6) Environments
7) Filters
8) Projects
9) Schedules
10) Selectors
11) Servers
12) Steps
13) SysConfig Parameters
14) Templates

Build Analysis:

The build analysis runs through the parent level artifacts and their associated environments. The logs are not gathered in this section to reduce the time it takes to analyze the builds at a high level. A complete list of all analysis done for build analysis is available in Appendix B.

1) Builds
    a. The top level build artifacts seen when browsing to the Jobs page through the user interface
2) Build Results
    a. The step results for builds
3) Build Environments
    a. Build environments associated with the top level build artifact only
4) Build Result Environments
    a. Build environments associated specifically with the step result artifacts. Please note the build level environment is not repeated here. You will only have a result build environment if an environment group is associated with the step.

Log Analysis:

The log analysis typically takes the longest amount of time to run. The majority of the data within a mature Build Forge environment is contained within build logs, or the BOM. A complete list of all analysis done for log analysis is available in Appendix C.

1) Custom BOM
    a. The custom BOM is created from adapters, or the .bom command
2) BOM Manifests
    a. The BOM manifest is automatically created by Build Forge and contains information such as which server the step ran on, as well as the step results for a build.
3) Build Logs
    a. The step logs for a build

Extra Options:

The extra options allow you to control how the RBFIPMT operates. There are six extra options available for use.

1) root
2) default
3) verbose
4) noclean
5) throttle
6) aggressivethrottle

Path:

This option changes the output path of the RBFIPMT. By default this path is wherever the current user path context is with the addition of \test on Windows or /test on UNIX. For example if the current path context on command line is C:\temp the root path for the RBFIPMT will be c:\temp\test.

Verbose:

This mode prints out the reports gathered onto the console screen and into the profiler.log file.

Throttle:

Throttle uses sleeps based on the file size of the currently processing profile artifact.

Aggressive Throttle:

This is used in conjunction with throttle and increases the sleeps more dramatically.

Usage Examples

The examples below assume the following:

1) The Build Forge host machine is the rbf-server
2) The port of the services layer is the default - 3966
3) The BF API user is profiler with the password of profiler
4) The Profiler.jar and rbf-services-client-java.jar exist in the same directory
5) The current path context is c:\temp

To run all tests:

```
java -jar Profiler.jar --user=profiler --pass=profiler --host=rbf-server --all
```

To run config tests only:

```
java -jar Profiler.jar --user=profiler --pass=profiler --host= rbf-server --config
```

To run all config and build tests:

```
java -jar Profiler.jar --user=profiler --pass=profiler --host= rbf-server –config --build
```

To run all config and build tests with the extra options verbose:

```
java -jar Profiler.jar --user=profiler --pass=profiler --host= rbf-server –config --build --verbose
```

To run all tests with the extra option throttle:

```
java -jar Profiler.jar --user=profiler --pass=profiler --host= rbf-server –config --build --throttle
```

Reports

Information Gathered

There are three types of information gathered for each artifact.

1) Simple Counters
    a. Simple counter comprise typically of totals – for example total projects profiled, total steps, etc. These simple counters are useful in providing clues for the parent artifacts for the various complex counters.
2) Group By Collections
    a. A group by collection groups instances of an artifact based on a commonality. In the Introduction there is an example of a group by collection which groups access groups by owning group. The parent artifact in the group by case is always the item the artifact is being grouped by. For example in the report Grouped By Group Owner the parent artifact is the owning group.
3) Complex Counters
    a. Complex counters store information upon which the statistical information is gathered such as average and standard deviations.

For the more complex static data points the following items are gathered:

1) Total
    a. Represents the total number of artifacts processed
2) Average
    a. Represents the average for the value gathered. In cases such as project this holds little meaning as it will always be one, however in cases such as steps this holds more meaning as there can be more than one step per project. The same holds true for min and max values.
3) Maximum Value
    a. The highest value found.
4) Minimum Value
    a. The lowest value found
5) Standard Deviation
    a. Represents the standard deviation calculation performed on the gathered artifacts. The closer to 0this number is the higher likelihood that most of the parent artifacts are close to the average.
6) Number of Parent Artifacts Processed
    a. Represents the total number of parent artifacts processed. This holds different meaning for different report types.

How to Read the Report

Presented here is a snippet from Access Groups with the three types of information gathered:

```
-----------------------------------------
Access Group Details
Date: Thu Jul 08 13:30:44 CDT 2010
-----------------------------------------
Access Group Count : 16
```

Grouped By Group Owner :
Total: 16.0
Average: 8.0
Maximum Value: 8.0
Minimum Value: 8.0
Standard Deviation: 0.0
Number of Parent Artifacts Processed: 2.0

User Group Memberships Per Access Group :
Total: 264.0
Average: 16.5
Maximum Value: 64.0
Minimum Value: 4.0
Standard Deviation: 18.411953
Number of Parent Artifacts Processed: 16.0

The first statistic is the simple count – how many access groups exist in this system. In this case there are 16.

The second statistic is a Group By. The total is again 16 as there were 16 access groups processed. The max and min values are both the same at 8. This states for this particular statistic there are only two owning groups in the current Build Forge system – each owning 8 groups. The standard deviation is 0 which further tells us that each parent artifact is equal to the average of 8. The parent artifacts in this group by example are the owning groups of which there are two.

The third statistic is the complex statistic type. This one is geared towards user memberships per access group. The total is listed at 264 which states there are 264 users in this BF system. He average is 16.5 which states on average there are 16.5 users in each access group. The max and min start to tell an interesting story – the max is 64 users in one access group, and min is 4 users in one access group. The standard deviation tells more of a story at 18.41 which states quite a few access groups do not have 16.5 members in them, rather are likely out of that range. The parent artifact in this case is represented by access groups of which there are 16.

Step Details
Date: Thu Jul 08 13:30:48 CDT 2010

Project Count : 370
Step Count : 4800

Step Command Size :
Total: 710087.0
Average: 147.9348
Maximum Value: 3387.0
Minimum Value: 3.0
Standard Deviation: 256.4545
Number of Parent Artifacts Processed: 4800.0

Threaded Steps Count :
Total: 415.0
Average: 1.1216216
Maximum Value: 17.0

Minimum Value: 1.0
Standard Deviation: 3.172318
Number of Parent Artifacts Processed: 370.0

Here we have a snippet from steps. The first two entries are simple counter variables – there are 370 projects and 4800 steps in this environment.

Further down are two complex statistics – one applied to individual steps as the parent artifact and one applied to projects as the parent artifact. The step command size illustrates the total amount of space used by step commands – 710,087 characters. On average there ar 147 characters per step, with the largest step containing 3387 characters and the smallest with only 3. The standard deviation of 256 tells us the step command sizes are all over the place and likely most are not the average of 147 characters.

For threaded steps the parent artifact is projects – 370 – and the goal of this report is to detail how many threaded steps exist per project. There are 415 threaded steps total out of 4800 steps. On average there is 1.12 threaded steps per project, with the most heavily threaded containing 17.

Appendix A – RBFIPMT Statistics Gathered for Configuration

Access Group Details

Simple counters:

Access Group Count

Complex counters:

Name	Parent Artifact
Default LDAP Access Groups (*)	Access Group
Subgroups Per Access Group	Access Group
User Group Memberships Per Access Group	Access Group

Group By:

Name	Parent Artifact
Grouped by Default	Default group\Non-Default Group
Grouped By LDAP DN	LDAP DN entry
Grouped By Group Owner	Owning Group

Adaptor Details

Simple counters:

Adaptor Count

Complex counters:

Name	Parent Artifact
Adaptor Size	Adaptor
Package Type Adaptors	Adaptor
Defect Type Adaptors	Adaptor
Source Type Adaptors	Adaptor
Test Type Adaptors	Adaptor

Group By:

Name	Parent Artifact
Grouped By Access Control	Access Group
Grouped By Adaptor Type	Adaptor Type

Adaptor Link Details

Simple counters:

Adaptor Link Count

Complex counters:

Name	Parent Artifact
Adaptor Link Environment	Adaptor Link
Active Adaptor Links	Adaptor Link

Debug Adaptor Links	Adaptor Link
Inactive Adaptor Links	Adaptor Link

Group By:

Name	Parent Artifact
Grouped By Adaptor	Adaptor

Build Class Details

Simple counters:

Build Class Count

Complex counters:

Name	Parent Artifact
Build Class Purge Count	Build Class
Build Class Purge Days	Build Class
Start On Entry	Build Class
Start On Exit	Build Class
Start On Purge	Build Class
Delete Console Data	Build Class
Delete Everything	Build Class
Delete Files Only	Build Class
Delete Logs Only	Build Class
Delete Logs and Files	Build Class
Purge Any Build	Build Class
Purge Keep 1 Passed	Build Class
Purge Only Failed	Build Class
Purge Only Passed	Build Class

Group By:

Name	Parent Artifact
Grouped By Access Control	Access Group
Grouped By Class Purge Rule	Purge Rule
Grouped By Start On Purge Project	Start On Purge Project
Grouped By Class Purge Type	Purge Type

Collector Details

Simple counters:

Collector Count

Complex counters:

Name	Parent Artifact
Collector Property Count	Collector
Built-in Type	Collector
Include Type	Collector
Run Command Type	Collector
Set Value Type	Collector

Group By:

Name	Parent Artifact
Grouped By Access Control	Access Group
Grouped By Collector Property	Collector Property Name

Environment Group Details

Simple counters:

Environment Group Count

Complex counters:

Name	Parent Artifact
Append	Environment Group
Clear	Environment Group
Hidden	Environment Group
Prepend	Environment Group
Set	Environment Group
Set if not Set	Environment Group
Unset	Environment Group
Environment Entry Count	Environment Group
Include Entry Type Count	Environment Group
Environment Variable Mode - Must Change	Environment Group
Environment Variable Mode – Normal	Environment Group
Environment Variable Mode - Read Only	Environment Group
Environment Variable Mode - Required	Environment Group
Environment Variable Mode - Suppress Display	Environment Group
Pull Down Entry Type Count	Environment Group
Pull Down Option Count	Pull Down Env Vars
Pull Down Option Parameter Size	Pull Down Env Vars
Pull Down Option Value Size	Pull Down Env Vars
Standard Entry Type Count	Pull Down Env Vars
Standard Entry Parameter Size - Individual variable basis	Env Var
Standard Entry Value Size - Individual variable basis	Env Var

Group By:

Name	Parent Artifact
Grouped By Access Control	Access Group

Filter Details

Simple counters:

Filter Count

Complex counters:

Name	Parent Artifact
Action Clear	Filter Group
Action Clear and Halt	Filter Group

Action Clear and Warning	Filter Group
Action Clear and Warning and Halt	Filter Group
Action Fail	Filter Group
Action Fail and Halt	Filter Group
Action Fail and Stop	Filter Group
Action Halt	Filter Group
Action Notify Changers	Filter Group
Action Pass and Stop	Filter Group
Action Warning and Stop	Filter Group
Action Warning	Filter Group
Filter Include	Filter Group

Group By:

Name	Parent Artifact
Grouped By Access Control	Access Group

Project Details

Simple counters:

Library Count
Project Count

Complex counters:

Name	Parent Artifact
Environment	Project
Fail Chain	Project
Library	Project
Pass Chain	Project
Project	Project
Sticky	Project

Group By:

Name	Parent Artifact
Grouped By Access Control	Access Group
Grouped By Build Class	Build Class
Grouped By Selector - no libraries	Selector

Schedule Details

Simple counters:

Schedule Count

Complex counters:

Name	Parent Artifact
Inactive Schedules	Schedule
Run Once Schedules	Schedule
Active Schedules	Schedule
Schedules Overriding Project's Build Class	Schedule
Class Purge Schedule	Schedule

Schedule Number of Variables	Schedule
Size of Environment Parameters	Schedule
Size of Environment Values	Schedule
Schedules Using Environment Sync	Schedule
Project Schedules	Schedule
Schedules Overriding Project's Selector	Schedule

Group By:

Name	Parent Artifact
Grouped By Access Control	Access Group
Grouped By Environment	Environment Group
Grouped By Project	Project
Grouped By Owner	User

Selector Details

Simple counters:

Selector Count

Complex counters:

Name	Parent Artifact
BF_NAME Selector Variables	Selector
Total Selector Property Count	Selector
Required	Selector
Include Property	Selector
Standard Property	Selector

Group By:

Name	Parent Artifact
Grouped By Access Control	Access Group
Grouped By Property Name	Selector Property Name

Step Details

Simple counters:

Project Count
Step Count

Complex counters:

Name	Parent Artifact
JPO Step Count	Project
Absolute Path Steps Count	Project
Environment groups Per Step Count	Project
Fail Chain Count	Project
Filter Count	Project
Inline Library Count	Project
Join Steps Count	Project
Pass Chain Count	Project
Relative Path Steps Count	Project

Selector Per Step Count	Project
Step Command Size	Step
Threaded Steps Count	Project

Group By:

Name	Parent Artifact
Steps Grouped By Access	Access Group
Grouped By Selector	Selector
Grouped By Directory Path	Step Path
Grouped By Environment	Environment Group
Grouped By Timeout	Step Timeout

SysConfig Details

Simple counters:

Auto Clean Audit Log Days
Auto Clean Error Log Days
Auto Clean Info Log Days
Auto Logoff Minutes
Auto Clean Warning Log Days
Cancel Check Frequency
Chain Tag Sync
Agent Connection timeout
Database Threshold Size
Enable Quick Start
Hard Run Limit
Max Console Processes
Max Purge Count
Max Manifest Refreshes
Run Queue Size
Max Server Tests
Purge Check Frequency
Server Max Retries
Step Max Retries
Terminate Thread

Template Details

Simple counters:

Notification Template Count

Complex counters:

Name	Parent Artifact
Template Body Size	Template Body
Unique Language Templates	Template
Step Notification Templates	Template
German Language Templates	Template Body
English Language Templates	Template Body
Spanish Language Templates	Template Body
French Language Templates	Template Body

Italian Language Templates	Template Body
Japanese Language Templates	Template Body
Korean Language Templates	Template Body
Portuguese Language Templates	Template Body
Simplified Chinese Language Templates	Template Body
Traditional Chinese Language Templates	Template Body
Template Type - Build Break	Template
Template Type - Build Start	Template
Template Type - Build Pass	Template
Template Type - Build Fail	Template
Template Type - Build Warn	Template
Template Type - Step Pass	Template
Template Type - Step Fail	Template
Template Type - Step Warn	Template
Template Type - Step Email	Template
Template Type - Purge Fail	Template
Template Type - System Alert	Template
Template Type - Filter Fire	Template
Template Type - Artifact Fire	Template

Group By:

Name	Parent Artifact
Template Type - Grouped By Project	Project
Template Type - Grouped By Type	Template Type

User Details

Simple counters:

User Count

Complex counters:

Name	Parent Artifact
Direct Group Memberships Per User	User
Ldap User Count	User
Max Builds Per User	User

Group By:

Name	Parent Artifact
Grouped By Domain	LDAP Domain
Grouped By Locale	User Locale
Grouped By Time Zone	User Time Zone

Appendix B – RBFIPMT Statistics Gathered for Build

Build Details

Simple counters:

Build Count

Complex counters:

Name	Parent Artifact
Build Result - Fail	Build
Build Result - None	Build
Build Result – Pass	Build
Build Result - Running	Build
Build Result - Warn Failed Step	Build
Build Result - Filter Match	Build
Build Duration	Build

Group By:

Name	Parent Artifact
Builds Grouped By Access Control	Access Group
Builds Grouped By Build Class	Build Class
Builds Grouped By Engine ID	BF Engine ID
Builds Grouped By Owner	User
Builds Grouped By Project	Project
Builds Grouped By Selector	Selector
Builds Grouped By State	Build State

Build Result Details

Simple counters:

Build Count
Step Result Count

Complex counters:

Name	Parent Artifact
Result Fail Chain Count	Build
Result Fail Wait Count	Build
Result Inline Count	Build
Result Pass Chain Count	Build
Result Pass Wait Count	Build
Build Result Command Size	Build Standard Step Result
Command Size for If portion of JPO step	Build JPO Step Result
Command Size for Else portion of JPO step	Build JPO Step Result
Result Condition Size	Build JPO Step Result
Build Result Duration	Build Step Result
Build Step Result – Cancelled	Build
Build Step Result - Failed With Warning	Build
Build Step Result – Failed	Build
Build Step Result - Filter Warning	Build

Build Step Result - None	Build
Build Step Result - Passed	Build
Build Step Result – Running	Build
Build Step Result – Skipped	Build
Build Step Result - Stopped	Build
Result Absolute Path Count	Build
Result Broadcast Count	Build
Results With Filter Count	Build
JPO IF Steps	Build
Build Result Type Standard	Build
JPO While Steps	Build

Group By:

Name	Parent Artifact
Results Grouped By Access Control	Access Group
Results Grouped By Build	Build
Results Grouped By Server	Server
Command Size Grouped By Build	Build
Result Duration Grouped By Build	Build

Build Environment Details

Simple counters:

Build Count
Build Environment Count

Complex counters:

Name	Parent Artifact
Build Environment Action - Append	
Build Environment Action - Clear	
Build Environment Action – Hidden	
Build Environment Action - Prepend	
Build Environment Action - Set	
Build Environment Action - Set If Not Set	
Build Environment Action - Unset	
Include Build Environment Variables	
Include Depth	
Build Environment Mode - Must Change	
Build Environment Mode - Normal	
Build Environment Mode - Read Only	
Build Environment Mode – Required	
Build Environment Mode - Suppress Display	
Origin type - Built In (BF_)	
Origin Type - Tag Variable	
Origin Type - User Defined Env Var	
Build Environment Parameter Name Size	
Build Environment Parameter Value Size	
Build Environment Pull Down Count	
Build Environment Pull Down Parameter Name Size	

Build Environment Pull Down Parameter Value Size	
Build Environment Standard Count	

Group By:

Name	Parent Artifact
Grouped By Origin User Type	
Grouped By Origin	
Parameter Name Size Grouped By Build	
Parameter Value Size Grouped By Build	
Grouped By Build Environment Mode	
Grouped By Build Environment Action	
Build Environment Count Grouped By Build	
Build Environment Count Grouped By Build	

Build Environment Result Details

Build Results Environment Details builds on Build Environment Details and collects the same information with the addition of the below statistics.

Simple counters:

Step Result Count

Group By:

Name	Parent Artifact
Step Result Count Grouped By Result	
Step Result Parameter Name Size Grouped By Result	
Step Result Parameter Value Size Grouped By Result	

Appendix C – RBFIPMT Statistics Gathered for Logs

Build BOM Details

Simple counters:

Build BOM Count
Build BOM Data Row Count
Build BOM Section Column Count
Build BOM Section Count

Complex counters:

Name	Parent Artifact
Build BOM Data Row Size	
Build BOM Section Name Size	
Build BOM Sections Count	
Columns Grouped By Section Name	BOM Section

Group By:

Name	Parent Artifact
Build BOM Grouped By Build	Build
Build BOM Section Columns Grouped By Build	Build
Build BOM Section Columns Grouped By Section	BOM Section
Build BOM Sections Grouped By Build	Build
Build BOM Sections Grouped By Name	Section Name

Build BOM Manifest Details

Simple counters:

BOM Manifest Count

Complex counters:

Name	Parent Artifact
Build BOM Manifest Name Size	
Build BOM Manifest Value Size	

Group By:

Name	Parent Artifact
BOM Manifest Count Grouped By Result	

Build Log Details

Simple counters:

Adaptor Count

Complex counters:

Name	Parent Artifact
Build Log Size	Step Result

Group By:

Name	Parent Artifact

Build Logs Grouped By Type	Log Type (ENV,SET,etc)
Build Log Count Grouped By Result	Step Result
Log Output Size Grouped By Build	Build

Appendix D – Usage Statement

USAGE: java -jar Profiler.jar {Connection Parameters} {Profile Mode} [Extra Options]

-h, --help
Displays this help text, then exits.

Connection Parameters:

Required Connection Parameters:
--user=<username> - Username for the Build Forge API.
--pass=<password> - Password for the Build Forge API.
--host=<hostname or ip address> - Hostname for the Build Forge API.

Optional Connection Parameters:
--port=<port number> - Port for the Build Forge API.
--domain=<BF logical domain name> - LDAP Domain as specified in Build Forge.

Profile Modes:

--config - Run only configuration data collection.
--build - Run only build data collection.
--logs - Run only log data collection.
--all - Run both configuration and build tests. - This will run all three profile modes and can be time consuming.

Extra Options:

--root=<directory path> - Directory to use as the profiler root path. Do not use a trailing slash "\" or "/"
The current working directory of Profiler.jar will be used by default
--default - Use default values for the Build Forge API connection. This can be used in place of the Connection parameters specified above.
Default Values:
User: root
Pass: root
Host: localhost
Port: 3966
--verbose - Turn on verbose mode
--noclean - Do not clean the root directory after all tests are run
--throttle - Throttle the speed using sleeps - Caution - this will slow the process down.
--aggressivethrottle - used in conjunction with --throttle. This will use longer sleeps

Examples:

To display help:
java -jar Profiler.jar -h

```
        java -jar Profiler.jar --help
    To use an alternate user\pass on the localhost:
        java -jar Profiler.jar --user=build --pass=build --all
    To use a non-default host:
        java -jar Profiler.jar --host=serverName --all
    To run all tests in verbose mode host:
        java -jar Profiler.jar --default --all --verbose
```

NOTE: The rbf-services-client-java.jar for your version of Build Forge MUST be in the same directory as the Profiler.jar file.

This file is typically found in
%BF_HOME%\webroot\public\clients\java on Windows or
$BF_HOME/webroot/public/clients/java on UNIX.
Alternatively you can go to http://server/clients/java to download the jar