



## BI Validator: Stress Test Parameters

*Prepared by*

**datagaps**

<http://www.datagaps.com>

<http://www.youtube.com/datagaps>

<http://www.twitter.com/datagaps>

*Contact*

**contact@datagaps.com**

Table of Contents

**Purpose of the DOCUMENT ..... 1**

**Setup Screen..... 2**

    Purge Parameters:.....2

        Purge before each run: .....2

        Ignore Cache.....2

        Do not Purge Cache.....2

    Ramp Up Time:.....2

    Think Time:.....2

**Result Parameters..... 4**

    Average Runtime.....4

    Average Initial Runtime .....4

    Minimum Runtime .....4

    Maximum Runtime .....4

    Number of Failures.....4

    Run Count .....4

    SLA Failures.....5

    Web Response Code.....5

    Web Response Details.....5

## PURPOSE OF THE DOCUMENT

This document explains the parameters that can be configured in BI Validator's Stress Test Plan Wizard and the parameters in the response.

## SETUP SCREEN

### Purge Parameters:

#### Purge before each run:

BI Validator submits an ODBC call {SAPurgeAllCache()} using the "Issue SQL" functionality in OBIEE to purge the cache. The BI Connection user will need to have the appropriate admin privilege to run this SAPurgeAllCache() function.

#### Ignore Cache

BI Validator issues "Go URLs" with a request parameter of Action=RefreshAll. This does not use the BI Server cache for each and every request. But unfortunately, when the Go URL has this parameter it is ignoring the parameters given for prompts

#### Do not Purge Cache

This leverages cache,

### Ramp Up Time:

Assume 30 concurrent users. The fastest ramp up that BI Validator can accommodate is 2 users for second. In a realistic environment, this is not how it happens. So, we want to have a way of specifying the ramp up time to simulate a realistic scenario. Lets say 30 users and you specify 180 seconds as Ramp Up Time. (every six seconds, a user is added). this happens across each of the runs.

### Think Time:

Objective is to simulate a realistic scenario. Let's say you specify 5 pages per visit (different param), between each page, this parameter can be used to specify the number of seconds to wait. So, let's us say that you specify a value of 30 seconds, BI Validator randomly generates a number between 0-30 (seconds) and based on the value, it hits the BI server after the interval.



## RESULT PARAMETERS

### Average Runtime

Measures the average total response time for each dashboard page during a run. The total response time includes is calculated after taking into consideration the time taken by the individual reports in a dashboard page

### Average Initial Runtime

Measures the average initial response time for each dashboard page during a run. Does not take the time taken by individual reports for consideration.

### Minimum Runtime

Shows the minimum total response time for each dashboard page during a run

### Maximum Runtime

Shows the maximum total response time for each dashboard page during a run

### Number of Failures

Counts the number of times the dashboard page request failed due to SLA Timeout or 403 web response or any other failure

### Run Count

Counts the number of times a dashboard page was requested as part of a run. BI Validator selects dashboard pages randomly from the pool of pages that are part of the test plan

## **SLA Failures**

Counts the number of times a dashboard page exceeded the SLA (Service Level of Agreement) time in a run

## **Web Response Code**

Shows a count of response codes received during a run. For example – 202 (ok) or 403 (Forbidden)

## **Web Response Details**

Shows the details of all the responses for a dashboard page during a run including the initial response time, report response time (docpart), content length, start time, end time, duration, user and the actual response received