

# High Availability Test Tool

## Version 2.1

January 10, 2018

PUBLIC

# High Availability Test Tool Version 2.1

## Introduction

- **Mission**

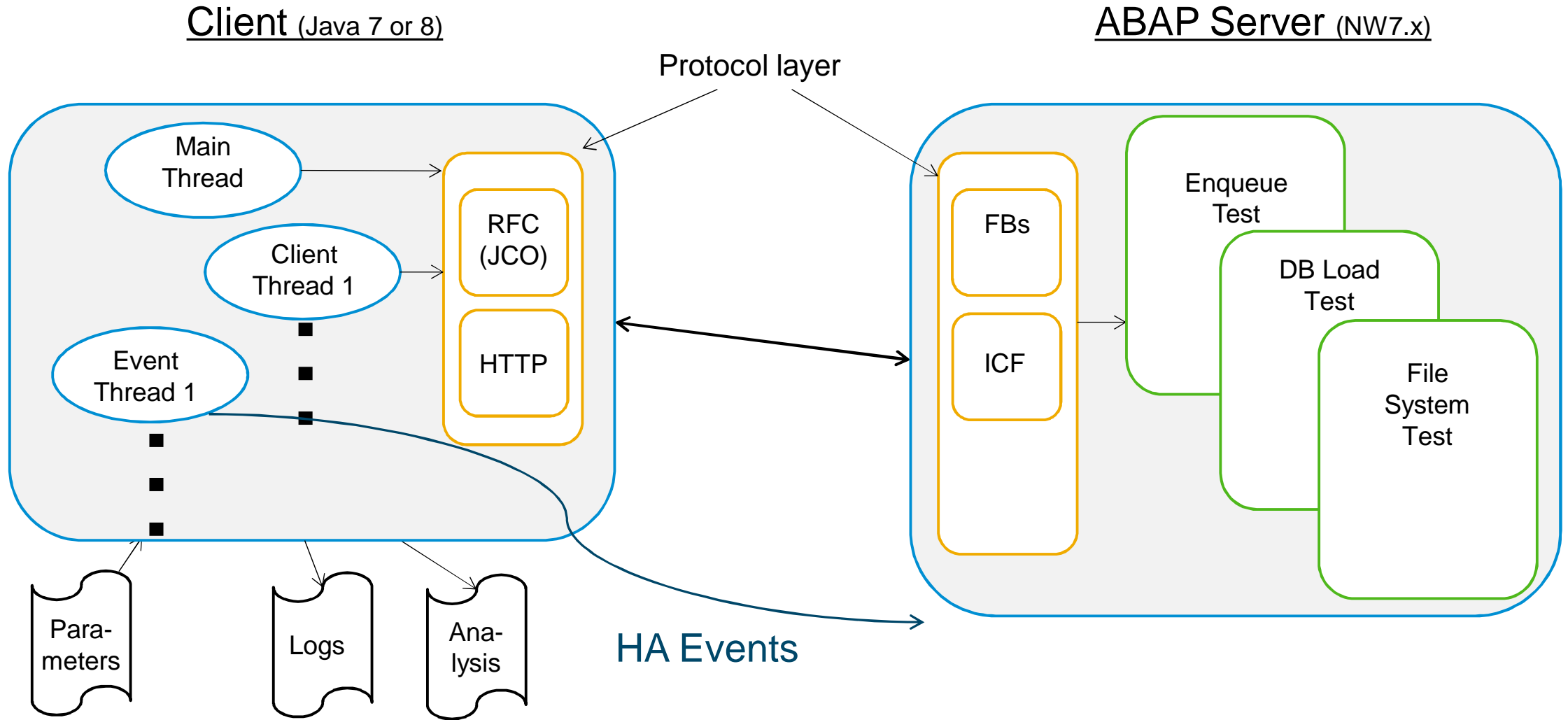
- HA tests for SAP AS ABAP 7.00 and higher versions

- **Key Characteristics of Version 2.1**

- Simple installation
- Automated test execution
- Quick test assessment
- Automated check against expected result
- Configurable and defined load
- Integrated fail-over event handling
- Dedicated test classes to cover single point of failures
- Coherent user-experience for all test cases
- Dedicated test tables and test data to avoid interference with SAP applications

# High Availability Test Tool Version 2.1

## Architecture



(HTTP protocol and file system test as of version 2.11)

# High Availability Test Tool Version 2.1

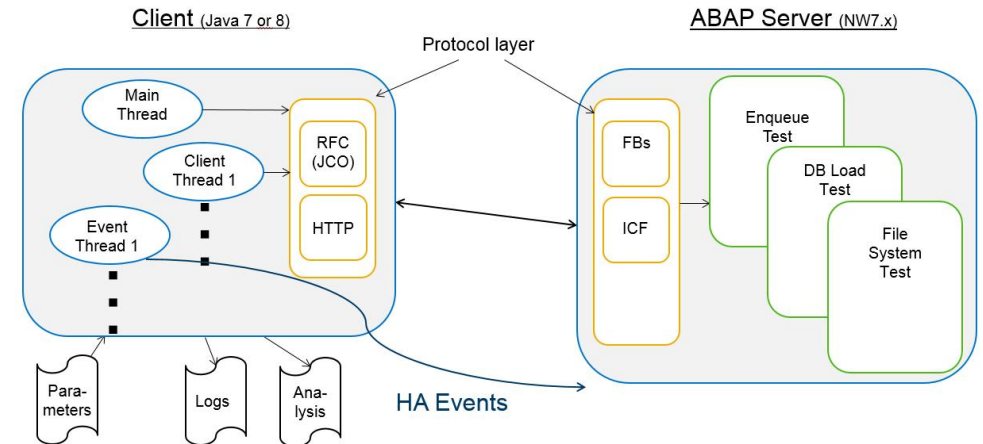
## Installation, Configuration and Execution

### ■ Server

- Prerequisite
  - WebAS ABAP 7.0 and higher versions
- Package
  - ABAP transport with programs, tables, and data
  - Customizing transport with role SAP\_BC\_HALOAD
- Configuration
  - Create user and assign role SAP\_BC\_HALOAD to him/her
  - Activate ICF service hatool (as of v2.11)

### ■ Client

- Prerequisite
  - JDK/JRE 7 or 8 (SAP JVM recommended)
  - SAP JCo 3.0
  - Graphics library d3.min.js (optional; as of v2.12)
- Package
  - Client driver (jar-file)
  - Property files (complete, event and quick start property file)
  - Template for checking against expected result
  - Template for HTML output (as of v2.12)
- Configuration
  - Adapt property file(s) and (optionally) check template file
- Start Test
  - `java -cp haTestTool.jar;C:\sapjco3\sapjco3.jar com.sap.test.haload.ClientDriver file=ha.properties`

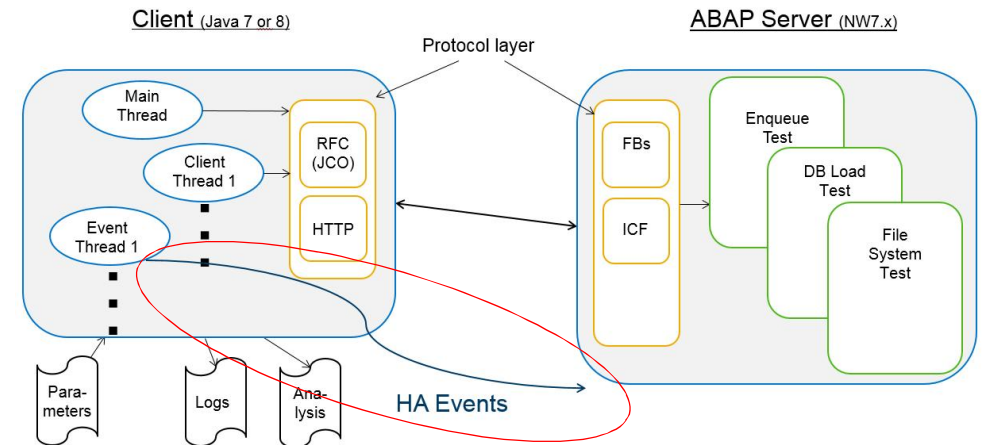


# High Availability Test Tool Version 2.1

## Automated System Fail-Over

### ■ HA Events

- Programs or scripts to be called on client side
  - Call mechanism for events
  - Collection of the event output
  - Events have to be scripted by end-user
- Single Events
  - Independent of other events
  - Start certain time after beginning of highload phase
- Event chains
  - First event starts certain time after beginning of highload phase
  - Other events start after their predecessor has finished
- Single Events combined with Event Chains
  - Any combination of single events and event chains are possible
- Event call parameters can be variables (as of v2.13)
  - Test properties and user-defined variables allowed



# High Availability Test Tool Version 2.1

## Customer Readiness

### ■ Security

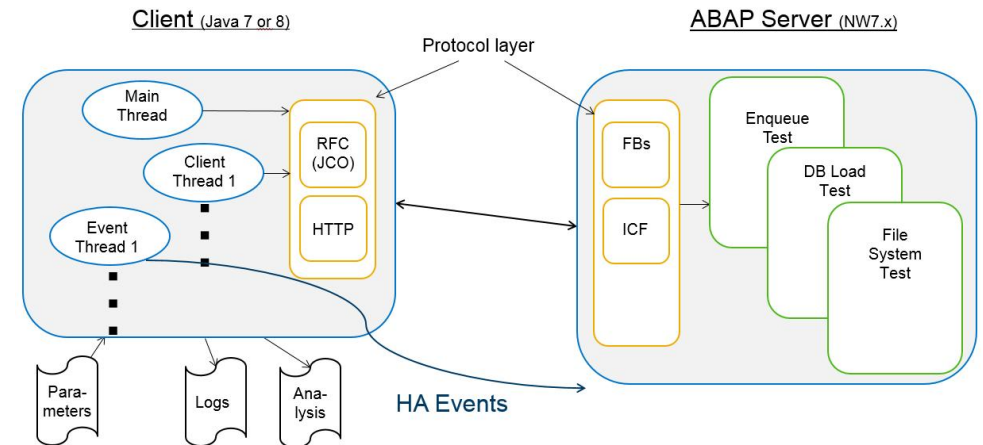
- SNC (Secure network Communication)
- HTTPS (as of v2.11)
- Authorization check on server + user role
- Fixed directory structure
- Uniqueness of object names on the server

### ■ Cleanup

- Uninstallation on the server via transport

### ■ Delivery Process

- Download the HA Test Tool package from the SAP Store
- See SAP note 2081226 for details



# High Availability Test Tool Version 2.1

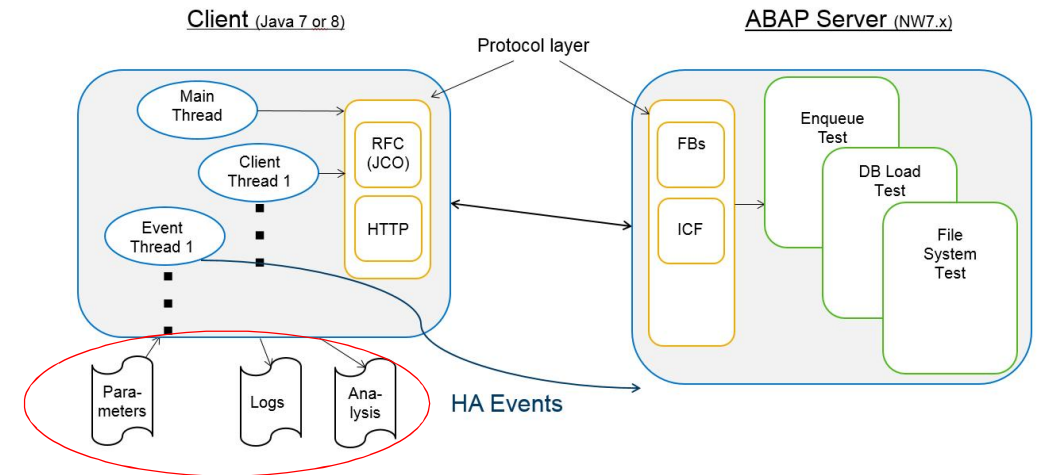
## Data Output

### Directories

- Test run specific output directory
- Data of former runs is not overwritten

### Data Files

- Dedicated output files per test run
  - Log file
  - Summary file (contains detailed analysis information)
    - Text file
    - XML file
    - HTML file (graphical; as of v2.12)
    - .csv file for performance data
- Dedicated output file per client thread
  - .res file (protocol of server calls)
  - .err file (extract of erroneous server calls)
  - .elg file (error message and stack trace for each erroneous server call)
  - .summary file
- Dedicated output file per HA event
  - .log file (log file of event thread of HA Test Tool)
  - .out file (redirection of stdout and stderr of event)

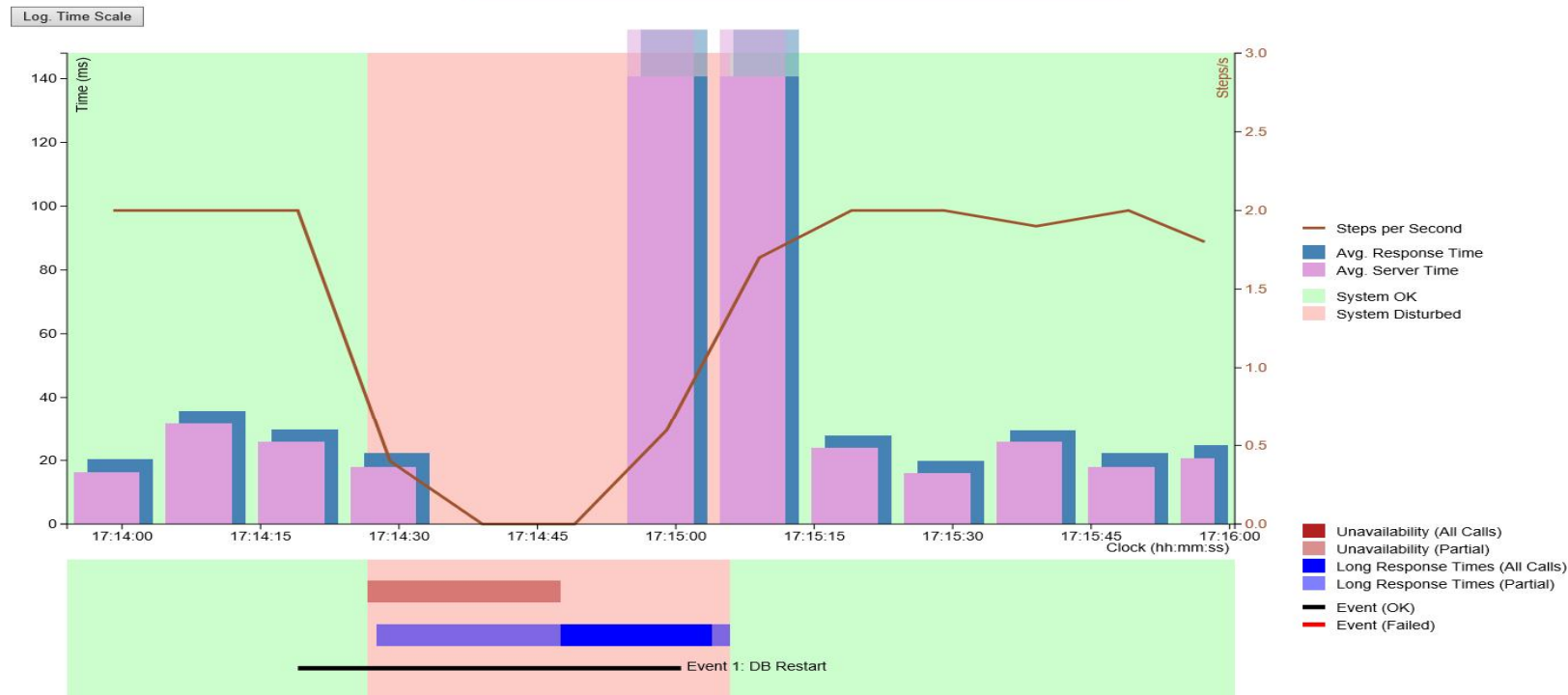


# High Availability Test Tool Version 2.12 and above

## Graphical Analysis of Test Run - Example of file loadtest.html

Graphics library d3.min.js required (is open source)

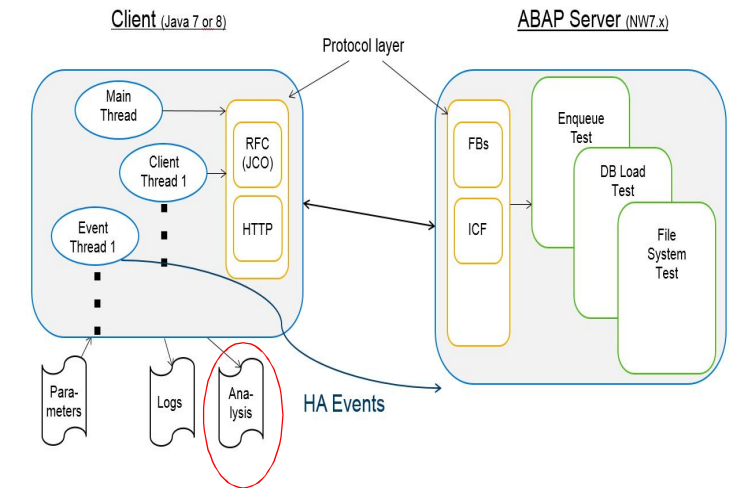
Summary of Test Run with ID: v2.12+check.db.restart.file



Exitcode: 0

+ Check against expected Result

+ Properties





# High Availability Test Tool Version 2.1

## Analysis of Test Run (I) - Example of file loadtest.summary

```
LOADTEST v2.13
Started at          2016-09-14 14:52:14.654

Test RunID:        213+c.DbRstFil
Test class:        CL_HA_DB_LOAD
Call type:         RFCStateful
Destination:       xxxxxxxx
Output directory:  C:\HATool\output\213+c.DbRstFil\xxxxxxx\RFCStateful\CL_HA_DB_LOAD
checkfile:         C:\HATool\check\db.restart.xml
Number of clients: 10
Steps per loop:   5
Loops per session: 1
Think time(s):    5
Subtest Parameters: ItemsPerOrder:50
Test duration(s): 300
Duration after events: 60
```

Internal ID of subtest: 3183

```
High load phase started at: 2016-09-14 14:53:07.413
High load phase ended at:   2016-09-14 14:55:15.264
2016-09-14 14:55:36.904 Test finished
```

### HA EVENTS

-----

```
Event 1: DB Restart
Status: EXECUTED (Exit code: 0)
Duration: 2016-09-14 14:53:32.451 - 2016-09-14 14:54:15.262 (42,811s)
```

General Test  
Information

HA Events

# High Availability Test Tool Version 2.1

## Analysis of Test Run (II) - Example of file loadtest.summary

### A V A I L A B I L I T Y and R E S P O N S I V E N E S S

---

System ok: 2016-09-14 14:52:17.500 - 2016-09-14 14:53:40.072  
System disturbed: 2016-09-14 14:53:40.073 - 2016-09-14 14:54:15.910 (35,837s)  
System ok: 2016-09-14 14:54:15.911 - 2016-09-14 14:55:34.733

The periods of disturbance of the system are caused either by erroneous calls that indicate that the system or parts of the system are not available or by calls that exceed the threshold for long response times suggesting that long wait times inside the system occurred during the calls.

For more details, see the following sections:

- Disturbance, indicated by erroneous calls
- Disturbance, indicated by long response times

### D I S T U R B A N C E, indicated by E R R O N E O U S C A L L S (U N A V A I L A B I L I T Y)

---

Severe errors in part of the calls: 2016-09-14 14:53:40.073 - 2016-09-14 14:54:00.884 (20,811s)

The errors that indicate periods of restricted or no availability are caused by following reasons:

Component(s) on the server could not be reached.

This is indicated by the error code APP\_NOTAVAIL\_ERR.

For more details, search for APP\_NOTAVAIL\_ERR in the client output files

### D I S T U R B A N C E, indicated by L O N G R E S P O N S E T I M E S

---

Threshold for long response times = 10 seconds

Partially long response times: 2016-09-14 14:53:41.035 - 2016-09-14 14:54:15.910 (34,875s)

Availability and  
Responsiveness of the  
system

Reasons for  
disturbance

Erroneous calls

Calls with long  
response times

# High Availability Test Tool Version 2.1

## Analysis of Test Run (III) - Example of file loadtest.summary

### VERIFICATION

Verification of server data after the test run showed no errors

Verification result of frame test class: CL\_HA\_TESTFRAME\_DB

OK: All DB tables have the correct number of rows with int\_runid 0.

Verification result of test class: CL\_HA\_DB\_LOAD

DB table size check of int\_runid 3183

OK. Number of rows in DB table HALOAD\_DB\_MAT = 1000

OK. Number of rows in DB table HALOAD\_DB\_CUST = 1000

OK. Number of rows in DB table HALOAD\_DB\_ORDER = 60

OK. Number of rows in DB table HALOAD\_DB\_ORDITM = 1500

Result of checks of int\_runid 3183

No check errors occurred

Verification summary of test class: CL\_HA\_DB\_LOAD

OK: All DB tables have the correct number of rows for checked int\_runids.

OK: No check errors occurred during the test.

### ERROR OVERVIEW

Number of erroneous calls: 3

Errors during execution of steps

Number of application errors (APP\_NOTAVAIL\_ERR): 3

Verification of server data after the test run

Verification made during the  
test run

Error summary

# High Availability Test Tool Version 2.1

## Analysis of Test Run (IV) - Example of file loadtest.summary

### PERFORMANCE and THROUGHPUT

```

Time range: 2016-09-14 14:53:07.413 - 2016-09-14 14:55:15.264
Number of steps: 184
Number of logins: 40
Steps per second: 1,4
Avg. response time in ms: 1429,7
Avg. elapsed time(Server) in ms: 32,9
Avg. response time of login in ms: 23,7
Step 1
  Avg. response time in ms: 3575,2
  Avg. elapsed time(Server) in ms: 40,9
Step 2
  Avg. response time in ms: 1789,4
  Avg. elapsed time(Server) in ms: 47,6
Step 3
  Avg. response time in ms: 31,1
  Avg. elapsed time(Server) in ms: 25,6
Step 4
  Avg. response time in ms: 30,2
  Avg. elapsed time(Server) in ms: 24,8
Step 5
  Avg. response time in ms: 1295,0
  Avg. elapsed time(Server) in ms: 24,4
  
```

Tact duration = 10 seconds

Number of Tact	Start of Tact	Number of Steps	Steps per Second	Avg. Response Time (ms)	Avg. Server Time (ms)	Avg. Login Time (ms)
1	2016-09-14 14:53:07.414	20	2,0	19,9	15,2	24,5
2	2016-09-14 14:53:17.414	20	2,0	33,8	27,6	0,0
3	2016-09-14 14:53:27.415	20	2,0	27,6	21,4	26,0
4	2016-09-14 14:53:37.417	4	0,4	19,0	13,0	26,5
5	2016-09-14 14:53:47.418	0	0,0	0,0	0,0	0,0
6	2016-09-14 14:53:57.420	0	0,0	0,0	0,0	0,0
7	2016-09-14 14:54:07.421	10	1,0	25804,1	195,4	20,0
8	2016-09-14 14:54:17.422	17	1,7	54,1	49,5	0,0
9	2016-09-14 14:54:27.423	20	2,0	27,2	21,9	23,0
10	2016-09-14 14:54:37.424	20	2,0	23,4	18,2	22,6
11	2016-09-14 14:54:47.425	20	2,0	30,7	25,2	24,0
12	2016-09-14 14:54:57.426	19	1,9	21,1	16,4	23,0
.....						

Response times and throughput of complete run

Granular analysis of response times and throughput

# High Availability Test Tool Version 2.1

## Analysis of Test Run (V) - Example of file loadtest.summary

C H E C K a g a i n s t E X P E C T E D R E S U L T  
-----

Check result against C:\HATool\check\db.restart.xml

Check for unavailability

OK: 1 period(s) of unavailability occurred; up to 1 expected.

OK: Duration of period 1: 20,811 seconds; up to 600 seconds acceptable.

Result:

OK: Period(s) of unavailability behaved as expected.

Check for long response times

OK: 1 period(s) of long response times occurred; up to 1 expected.

OK: Duration of period 1: 34,875 seconds; up to 600 seconds acceptable.

Result:

OK: Period(s) of long response times behaved as expected.

Check for errors

OK: Errors of category APP\_NOTAVAIL\_ERR expected.

Result:

OK: Only expected errors occurred.

Final result:

OK: Outcome of test run was as expected.

E X I T C O D E  
-----

Exit code based on the analysis of the test run: 0 (OK)

Check against expected result

Exit code of test run