Automated Performance Testing of SAP Process Orchestration

Git REPOSITORY TEST DOCUMENT

INT603

Exercises / Solutions  
Nicholas Holshouser / SAP Labs, LLC

Abhinag Palvadi / SAP Labs India Pvt. Ltd.

Andre Bakofen / SAP AG

**TABLE OF CONTENTS**

[1 Introduction 3](#_Toc459188748)

[2 Automated Testing 3](#_Toc459188749)

[3 Testing as a Service 3](#_Toc459188750)

# **Introduction**

This document is the master test document for TechEd CodeJam Session INT603. For automated tests the instructions are also maintained in the repository. This ensures all tests are run using the correct documents, tools, test plans, and reporting templates.

# Automated Testing

* Run the test as described in this document.
* If you find any problems or discrepancies please make your corrections and resubmit so these can be reviewed and (as needed) incorporated into the master branch

# Testing as a Service

It’s possible to automate even the infrastructure but in our case we only have an hour so the backend test systems are already provisioned and available. Based on your local laptop image name please pick your backend from the list below and use those configuration details.

Access Details for Backend

|  |  |  |
| --- | --- | --- |
| BACKEND SYSTEM | IP ADDRESS | CERTIFICATE |
| INT603\_BACKEND01 | <IP> | INT603\_BACKEND01.ppk |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Access to the backend is using ‘root’ only and by certificate only

root: root user is accessible using certificate login only

The root user certificate is available in the repository

You can login to the backend (not necessary but possible) using Putty and your root certificate.

| **Explanation** | **Screenshot** |
| --- | --- |
| 1. Unzip Jmeter 3.0 |  |
| 1. Open Jmeter using jmeter.bat file |  |
| 1. Check your backend system is running and available:   http://<yourbackendIP>/nwa  User: Administrator  Password: Abcd1234 |  |
|  |  |
| 1. Load the initial connection test plan in JMeter   Test Plan:  D:\Files\Session\INT603\test603\jmeter30\scripts\INT603\_ConnectionTest.jmx |  |
| 1. Open the test plan and navigate to the User Defined Variables element and modify configuration for your test system by changing inputing the IP address |  |
| 1. Run the initial connection test to ensure you are connected to your backend.   In the View Results Tree you should see a green success icon  and you can view the raw request details and the response.  The test simply gets the WSDL of the deployed Integration Flow which we will use as the test target. |  |
| 1. Start backend performance monitor agent using Git Bash   JMeter uses a plug-in that connects to a running ServerAgent on the backend system and gets system performance metrics during the test. | Start Git Bash directly from the Git GUI. This has the advantage of putting you into the right directory ☺    Or from Start => Programs |
| 1. We use the program plink.exe (from the Putty package) to start the backend performance agent. The is a copy of plink.exe in your repository scripts directory   Plink must be run from a command prompt such as Git bash. |  |
| 1. Check connection to the backend using the command:   CERTS/PLINK.EXE root@<your ip address> -i CERTS/INT603\_BACKEND##.ppk 'df -h' |  |
| 1. Start the perfmon agent   CERTS/PLINK.EXE –I CERTS/INT603\_BACKEND##.ppk root@<your ip address> -i 'bash /hana/shared/agent/startAgent'' |  |
| 1. Start backend detailed performance monitoring (nmon)   This step is optional – you can also take more detailed performance metrics using tools such as nmon (UNIX/LINUX) | OPTIONAL |
| 1. Run the warm-up test |  |
| 1. Check the results – there should be no errors |  |
| 1. Open NWA and check the SOA monitoring on backend |  |
| 1. Open PO Performance Monitors (in NWA) and check the detailed performance views |  |
|  |  |
|  |  |

Appendix

We can also look at SYSTEMS as part of the repository and as we develop the system we can update master and then test the new feature

SYSTEMS can be standardized and then created dynamically in AWS. This includes the test client itself.

Take a vanilla system (Windows, Ubuntu, Linux) and make base test client

Test a standardized image of the application system (in our case it is a master image of a configured NW 7.5 Process Orchestration system with B2B and deployed content/scenario)

The standardized image can also be based on AWS and built dynamically using tools such as Cloud Foundry or Vagrant/Chef and EC2 API

Provide examples

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices.

