Polarion Diagnostic Tool (PDT) Handbook

1 How to Install and Run from Command-line

1. Unzip `com.polarion.diagtool.zip` to some temporary directory.
2. Make the folder `com.polarion.diagtool` the current working directory.
3. Edit `diagtool.properties` and adjust the properties if needed - read the comments in the file.
4. Execute `run.bat` (Windows) or `run.sh` (Linux) and wait until it finishes.
5. Results will be sent if configured to do so.

2 How to Run from the Polarion UI

Info: Polarion version 2013-SR2 or newer is required.

1. Configure the following scheduled job, if not already configured: (*-sendResults* must be set to "yes" or "no"):

```
<job name="Polarion Diagnostic Tool" id="execute.command" scope="system">
 <command>$[com.polarion.home]/diagtool/run$[shellExtension]</command>
 <arguments>
   <argument>-sendResults</argument>
   <argument>no</argument>
   <argument>-resultsDir</argument>
   <argument>$[jobDir]</argument>

   <!--
   <argument>-mail.subject</argument>
   <argument>MAIL_SUBJECT</argument>

   <argument>-mail.sender</argument>
   <argument>VALID_RETURN_EMAIL_ADDRESS</argument>

   <argument>-mail.recipient</argument>
   <argument>VALID_MAIL_RECIPIENT</argument>

   <argument>-mail.host</argument>
   <argument>MAIL_HOST</argument>

   <argument>-mail.port</argument>
   <argument>MAIL_SERVER_HOST_PORT</argument>

   <argument>-mail.protocol</argument>
   <argument>MAIL_SERVER_PROTOCOL</argument>

   <argument>-mail.attachmentThreshold</argument>
   <argument>MAIL_MAX_ATTACHMENT_SIZE_IN_BYTES</argument>

   <argument>-mail.auth</argument>
   <argument>MAIL_SERVER_AUTHENTICATION_REQUIRED_TRUE_OR_FALSE</argument>

   <argument>-mail.user</argument>
   <argument>MAIL_SERVER_USER</argument>

   <argument>-mail.password</argument>
   <argument>MAIL_SERVER_USER_PASSWORD</argument>
   -->
 </arguments>
</job>
```

1. Run the job from Monitor.
2. Results will be sent to a predefined email or FTP if *-sendResults* is set to "yes".

3 Command-line Configuration

Argument name	Description
-polarionHome	Polarion installation location (specify if PDT is not able to find it)
-resultsDir	Results directory (by default the *results* directory under the PDT installation root directory)
-configuration	Configuration file location (by default `diagtool.properties` in the PDT installation directory)

Additionally, anything which can be specified in the configuration file might be specified on the command line.
For example: **`-sendResults ask`** on the command line instead of **`sendResults = ask`** in the configuration file.

Command-line takes precedence over the configuration file.

4 Configuration File

Configuration property	Description
modules	Modules to run (`com.polarion.diagtool.core.EnvironmentCheck` and `com.polarion.diagtool.core.PolarionCheck` are run by default)
packedFileSizeThreshold	Maximum size of packed file (otherwise a warning is emitted)
sendResults	Controls whether or not to send the diagnostic results
mail.*	Mail settings
ftp.*	FTP settings

See bundled `diagtool.properties` for more information, including default values.

5 Results

Results are stored in a uniquely-named folder under the *results* directory. It always includes `log4j.txt`, which is a full log of the execution (includes all the errors and warnings), and `probes.properties` containing a recording of all probes in machine-readable format.

This might also include other packed files and directories. After the results are recorded, the results directory is packed as ZIP file whose name includes the local host name to identify it uniquely between machines.

6 Default Probes

Probe ID	Description
build.id	PDT build id
configuration.args	Command-line arguments
configuration.home	PDT installation location
configuration.file	Name of PDT configuration file
configuration.polarionHome	Polarion installation location
configuration.resultsParentDir	Results directory
configuration.packedFileSizeThreshold	Packed files size threshold
configuration.modules	Modules to run
configuration.*	Other configuration properties

7 Default module com.polarion.diagtool.core.EnvironmentCheck

7.1 Probes based on JVM results

Probe ID	Description
environment.jvm.systemProperty.*	JVM system properties
environment.jvm.os.environmentVariable.*	Environment variables
environment.jvm.os.name	OS name
environment.jvm.os.arch	OS architecture
environment.jvm.os.version	OS version
environment.jvm.availableProcessors	Number of available processors
environment.jvm.totalPhysicalMemorySize	Total physical memory size
environment.jvm.disk.*.totalSpace	Total disk space
environment.jvm.disk.*.freeSpace	Free disk space
environment.jvm.disk.*.usableSpace	Usable disk space
environment.jvm.localhost	Local host name and address

7.2 Windows-only probes

Probe ID	Description
environment.windows.wmic.OS	Operating system information (output of "wmic OS" call)
environment.windows.wmic.COMPUTERSYSTEM	Computer system information (output of "wmic COMPUTERSYSTEM" call)
environment.windows.wmic.LOGICALDISK	Local storage device information (output of "wmic LOGICALDISK" call)

7.3 Linux-only probes

Probe ID	Description
environment.linux.uname	Output of "uname -a" call
environment.linux.distribution.name	Distribution name
environment.linux.distribution.release	Distribution release information
environment./proc/cpuinfo	Content of /proc/cpuinfo
environment./proc/meminfo	Content of /proc/meminfo

7.4 Apache-related probes

Probe ID	Description
environment.apache.*	Information about all found Apache HTTP server binaries

7.5 Subversion-related probes

Probe ID	Description
environment.subversion.*	Version of all found Subversion modules

7.6 PostgreSQL-related probes

Probe ID	Description
environment.postgresql.version.*	Version of found PostgreSQL DB for all found 'psql' in /usr/

8 Default module com.polarion.diagtool.core.PolarionCheck

Probe ID	Description
polarion.installed	Whether Polarion is installed (other probes are not recorded if Polarion is not installed)
polarion.home	Polarion installation location
polarion.*.eclipseproduct	Content of .eclipseproduct
polarion.*.polarion.ini	Packed file polarion/polarion.ini (Windows-only)
polarion.*.config.sh	Packed file etc/config.sh (Linux-only)
polarion.*.polarion.log	Packed file /var/log/polarion/polarion.log (Linux-only)
polarion.*.configuration	Packed directory polarion/configuration
polarion.*.etc	Packed directory etc (Linux-only)
polarion.*.license	Packed directory polarion/license
polarion.*.plugins	Directory listing of polarion/plugins
polarion.logs.jobs.*	Packed log files from all jobs from all local Polarion instances
polarion.logs.metadata.*	Packed .log files from all local Polarion instances
polarion.logs.*	Packed console.log, main logs, gc logs and RPC logs from all Polarion instances (packs only latest normal and latest reindex logs, incl. rotated files)
polarion.apache.config.*	Packed all Apache HTTP server config files
polarion.apache.logs.*	Packed all log files from Apache HTTP server (packs only latest file from all log categories)
polarion.svn.access.*	Packed access files from all local Subversion repositories
polarion.*.postgresql.conf	Packed file polarion/data/postgres-data/postgresql.conf
polarion.logs.postgresql.*	Packed latest postgresql .log file from all local Polarion instances

9 Optional module com.polarion.fieldtest.FieldTest

Info: This module is a replacement for former Polarion Field Test Suite.

9.1 Configuration

Configuration is stored in `fieldtest.properties` inside the PDT installation directory.

9.1.1 Configuring Which Tests to Execute

Configuration property `suites` defines which tests should be executed and in which order.

Tests measuring performance of a single use case while simulating concurrent accesses to the used resource ("load tests") can be disabled with the configuration property `disableLoadTests` set to `true` (disabled tests will still be present in the results, but with zero time).

9.1.2 Configuration of LdapSearchTest

Steps to convert Apache's mod_authnz_ldap configuration into `ldap.*` properties (if there are quote marks (") around the values, then those should not be put into configuration file):

- `AuthLDAPBindDN` is directly `ldap.bindDN`
- `AuthLDAPBindPassword` is directly `ldap.bindPassword`
- `AuthLDAPURL` is split into `ldap.host`, `ldap.usersDN` and `ldap.searchFilter`

Pattern for splitting the `AuthLDAPURL` (NAME is existing user name):

```
AuthLDAPURL "ldap://host:port/basedn?attribute?scope?filter"

ldap.host = ldap://host:port
ldap.usersDN = basedn
ldap.searchFilter = (attribute=NAME)
```

9.1.3 Configuration of SvnLogTest

Configuration property `svn.logLimit` defines how many revisions are requested during the test using one svn log command. Recommended value is 10000, but 1000 should be enough to show patterns in the results in case 10000 makes the test run for way too long.

9.1.4 Configuration of StressFileLockingTest

Configuration property `fileLocking.delay` defines delay between lock tries, `fileLocking.tries` defines how many times file will be locked, `fileLocking.path` defines path to the file, default is `POLARION_HOME/../shared/locking_test`.

9.2 Contained Tests

9.2.1 LdapSearchTest

Suite name	`com.polarion.fieldtest.ldap.LdapSearchTest`
Performs changes on the tested system	No
Stressed subsystems	Network connection to the LDAP server and LDAP server
Expected duration	<2 minutes
Useful for comparison	Yes if the search filters have the same complexity

Measures LDAP search performance and its dependency on concurrent access.

9.2.2 SvnLogTest

Suite name	`com.polarion.fieldtest.subversion.SvnLogTest`
Performs changes on the tested system	No
Stressed subsystems	Network connection to the Subversion repository and Apache containing Subversion repository as well as repository itself
Expected duration	<1 hour
Useful for comparison	Yes if the repositories have the same content

Measures Subversion's revision log performance and its dependency on concurrent read access.

9.2.3 Base64PerformanceTest

Suite name	`com.polarion.fieldtest.system.Base64PerformanceTest`
Performs changes on the tested system	No
Stressed subsystems	CPU and memory
Expected duration	~1 minute
Useful for comparison	Yes

Measures computing power by computing Base64 representation of pseudorandom streams of data.

9.2.4 DiskPerformanceTest

Suite name	`com.polarion.fieldtest.system.DiskPerformanceTest`
Performs changes on the tested system	Temporary files are created and deleted in the Polarion's data folder
Stressed subsystems	Disk
Expected duration	~10 seconds

Useful for comparison	Yes

Creates, overwrites and deletes temporary files in the `data` folder of the Polarion installation.

9.2.5 StressFileLockingTest

Suite name	`com.polarion.fieldtest.system.StressFileLockingTest`
Performs changes on the tested system	Temporary file is created if not exists.
Stressed subsystems	Disk
Expected duration	~5 seconds (with default configuration)
Useful for comparison	Yes

Measures file locking, how long it takes to lock file in shared folder for X times with Y delay and its average.

9.3 Probes

Probe ID	Description
fieldtest.configuration.*	Content of configuration file
fieldtest.polarion.configuration.*	Content of Polarion configuration file
fieldtest.time.*	Measured time of every test

9.4 Referential Results

*- times in the table below are in **seconds***

almdemo.polarion.com

Linux x64, Subversion 1.6, 1000 revisions

babybear.labs.polarion.com

Linux x64, Subversion 1.9, 1000 revisions

cns.labs.polarion.com

Linux x64, Subversion 1.9, 1000 revisions

Windows AWS Cluster

Windows Server 2012 x64, Subversion 1.9, 1000 revisions

Probe ID	almdemo.polarion.com	babybear.labs.polarion.com	cns.labs.polarion.com	Windows AWS Cluster
fieldtest.time.com.polarion.fieldtest.system.Base64PerformanceTest.testStreams	2.506	1.517	3.576	2.457
fieldtest.time.com.polarion.fieldtest.system.DiskPerformanceTest.testSmall	0.739	0.312	0.464	1.508
fieldtest.time.com.polarion.fieldtest.system.DiskPerformanceTest.testLarge	0.326	0.274	0.383	0.867
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearch		0.013	0.009	
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearchConcurrent1ThreadsWith10MillisDelay		0.018	0.011	
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearchConcurrent10ThreadsWith10MillisDelay		0.035	0.280	
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearchConcurrent20ThreadsWith10MillisDelay		0.067	0.211	
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearchConcurrent30ThreadsWith10MillisDelay		0.098	0.087	
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearchConcurrent1ThreadsWith30MillisDelay		0.013	0.012	
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearchConcurrent10ThreadsWith30MillisDelay		0.032	0.052	
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearchConcurrent20ThreadsWith30MillisDelay		0.076	0.060	
fieldtest.time.com.polarion.fieldtest.ldap.LdapSearchTest.testSearchConcurrent30ThreadsWith30MillisDelay		0.114	0.112	
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWithNoThreads	7.990	0.561	0.576	2.257
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWith1ThreadWith1SecDelay	8.193	0.570	0.592	2.205
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWith10ThreadsWith1SecDelay	11.069	0.946	0.677	2.600
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWith20ThreadsWith1SecDelay	14.174	0.736	0.808	3.326
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWith30ThreadsWith1SecDelay	16.450	0.734	0.820	4.064
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWith1ThreadWith10SecDelay	8.044	0.572	0.640	2.206
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWith10ThreadsWith10SecDelay	8.598	0.608	0.664	2.302
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWith20ThreadsWith10SecDelay	9.441	0.869	1.023	2.519
fieldtest.time.com.polarion.fieldtest.subversion.SvnLogTest.testLogWith30ThreadsWith10SecDelay	10.171	1.038	1.170	2.665

9.4.1 Internal Polarion Cluster

Linux x64, Linux NFSv4, fileLocking.delay = 0, fileLocking.tries = 10000

Probe ID	Time
fieldtest.time.com.polarion.fieldtest.system.StressFileLockingTest.testLock	6.806

10 In-Polarion Test Script

Extension Polarion Scripting is required (with the scripting servlet included) for in-Polarion performance test (so-called "In-Polarion Test Script"). Enter content of `scripts/PerfTest.groovy` into text field on `/polarion/scripting` URL, adjust the variables at the top and then run the script. Log from the script contains all the information needed to analyze the in-Polarion test results.

Performs changes on the tested system	No
Stressed subsystems	All of them, incl. Polarion
Expected duration	Depends on the used document
Useful for comparison	Yes if the used document is the same

Renders and exports document specified by variables projectId, spaceName and documentName.

11 File Locking Test Utility

File Locking Test Utility is an extended version of StressFileLockingTest which can be executed on its own. It is executed in a same way as Polarion Diagnostic Tool, but has different command-line arguments: `-locktest COMMAND FILE DELAY TRIES`.

Supported commands are:

- `lock`: locks the FILE
- `check`: checks if the FILE is locked
- `stress`: locks the FILE repeatedly
    - waits for DELAY milliseconds between attempts or not at all if DELAY is 0 (which is the default value)
    - will repeat TRIES number of times or forever if TRIES is 0 (which is the default value)

12 Troubleshooting

12.1 '"java.exe"' is not recognized as an internal or external command, operable program or batch file.

Java executable was not found. Set environment variable JAVA_HOME to JRE installation folder, or JDK_HOME to JDK installation folder, or JAVA_CMD to Java executable. Then reexecute PDT.

12.2 '"C:\Program' is not recognized as an internal or external command, operable program or batch file.

Environment variable JAVA_HOME, JDK_HOME or JAVA_CMD contains quotation marks. Remove them and reexecute PDT.

12.3 run.sh: line 57: java: command not found

Java executable was not found. Set environment variable JAVA_HOME to JRE installation directory, or JDK_HOME to JDK installation directory, or JAVA_CMD to Java executable. Then reexecute PDT.

12.4 Exception in thread "main" java.lang.UnsupportedClassVersionError: org/codehaus/groovy/tools/GroovyStarter (Unsupported major.minor version 49.0)

Used Java is too old. Java 6 or higher is required.