



# Development Testing Platform Engine for Java ユーザーズ ガイド

Version 10.3

Parasoft Corporation  
101 E. Huntington Drive, 2nd Floor  
Monrovia, CA 91016  
Phone: (888) 305-0041  
Fax: (626) 305-9048  
E-mail: [info@parasoft.com](mailto:info@parasoft.com)  
URL: [www.parasoft.com](http://www.parasoft.com)

# PARASOFT END USER LICENSE AGREEMENT

PLEASE READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") CAREFULLY BEFORE USING THE SOFTWARE. PARASOFT CORPORATION ("PARASOFT") IS WILLING TO LICENSE THE SOFTWARE TO YOU, AS AN INDIVIDUAL OR COMPANY THAT WILL BE USING THE SOFTWARE ("YOU" OR "YOUR") ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS OF THIS AGREEMENT. THIS IS A LEGALLY ENFORCEABLE CONTRACT BETWEEN YOU AND PARASOFT. BY CLICKING THE "ACCEPT" OR "YES" BUTTON, OR OTHERWISE INDICATING ASSENT ELECTRONICALLY, OR BY INSTALLING THE SOFTWARE, YOU AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND ALSO AGREE THAT IS IT ENFORCEABLE LIKE ANY WRITTEN AND NEGOTIATED AGREEMENT SIGNED BY YOU. IF YOU DO NOT AGREE TO THESE TERMS AND CONDITIONS, CLICK THE "I DO NOT ACCEPT" OR "NO" BUTTON AND MAKE NO FURTHER USE OF THE SOFTWARE.

## 1. DEFINITIONS

- 1.1. "**Concurrent User**" means a person that has accessed the Software at any given point in time, either directly or through an application.
- 1.2. "**Instance**" means a single occurrence of initialization or execution of software on one machine.
- 1.3. "**Licensed Capacity**" means the capacity-based license pricing metrics identified in the Order Instrument, including, without limitation, Concurrent Users, Node Locked machines, and Instances.
- 1.4. "**Maintenance**" means the maintenance and technical support services for the Software identified in the Order Instrument and provided by Parasoft pursuant to this Agreement.
- 1.5. "**Node Locked**" means a license for a single machine that has been authorized to run a single Instance of the licensed Software. A Node Locked license requires that users are physically present and not accessing the machine and using the Software from a remote location.
- 1.6. "**Software**" means Parasoft's software products, in object code form, that are commercially available at the time of Your order and identified on the Order Instrument, and any modifications, corrections and updates provided by Parasoft in connection with Maintenance.
- 1.7. "**Territory**" means the country or countries in which You have a license to use the Software, as specified in Your order for the Software; or, if no Territory is specified, the country from which Your order has been issued.
- 1.8. "**User Documentation**" means the user's guide, installation guides, and/or on-line documentation applicable to the Software. User Documentation does not include marketing materials or responses to requests for proposals.

## 2. GRANT OF LICENSE AND USE OF SOFTWARE

- 2.1. **License Grant.** Subject to the terms and conditions of this Agreement, Parasoft grants to You a perpetual, non-exclusive license to use the Software within the Territory, in accordance with the User Documentation and in compliance with the authorized Licensed Capacity. You acknowledge and agree that this Agreement only grants a license to the Software as set forth herein and does not constitute a sale of the Software by Parasoft. You have no right to resell the Software, whether by contract or by operation of applicable copyright law.

- 2.2. **Usage Rights.** You may only use the Software and/or the User Documentation for Your internal business operations and to process Your data. You shall not (a) permit any third parties or non-licensed entities to use the Software or the User Documentation; (b) process or permit to be processed any data that is not Your data; (c) use the Software in the operation of a service bureau; (d) sublicense, rent, or lease the Software or the User Documentation to a third party; or (e) perform, publish, or release to any third parties any benchmarks or other comparisons regarding the Software or User Documentation. You shall not make simultaneous use of the Software on multiple, partitioned, or virtual computers without first procuring an appropriate number of licenses from Parasoft. You shall not permit a third party outsourcer to use the Software to process data on Your behalf without Parasoft's prior written consent.
- 2.3. **License Keys.** You acknowledge that the Software contains one or more license keys that will enable the functionality of the Software and third party software embedded in or distributed with the Software. You may only access and use the Software with license keys issued by Parasoft, and shall not attempt to modify, tamper with, reverse engineer, reverse compile, or disassemble any license key. If Parasoft issues a new license key for the Software, You shall not use the previous license key to enable the Software.
- 2.4. **Archival Copies.** You may make one copy of the Software for back-up and archival purposes only. You may make a reasonable number of copies of the User Documentation for Your internal use. All copies of Software and User Documentation must include all copyright and similar proprietary notices appearing on or in the originals. Copies of the Software may be stored offsite provided that all persons having access to the Software are subject to Your obligations under this Agreement and You take reasonable precautions to ensure compliance with these obligations. Parasoft reserves the right to revoke permission to reproduce copyrighted and proprietary material if Parasoft reasonably believes that You have failed to comply with its obligations hereunder.
- 2.5. **Licensed Capacity.** Parasoft licenses Software based on Licensed Capacity for different types of usage, including, without limitation, Concurrent Users and Node Locked machines. A Concurrent User license allows multiple Concurrent Users to share access to and use the Software, provided that the number of Concurrent Users accessing the Software at any time does not exceed the total number of licensed Concurrent Users. A Node Locked license allows a single specified machine to run a single Instance of the Software. If an application accessing the Software is a multiplexing, database, or web portal application that permits users of such application to access the Software or data processed by the Software, a separate Concurrent User license will be required for each Concurrent User of such application. Regardless of usage type, You shall immediately notify Parasoft in writing of any increase in use beyond the Licensed Capacity. You must obtain a license for any increase in Licensed Capacity, and You agree to pay to Parasoft additional Software license fees, which will be based on Parasoft's then-current list price.
- 2.6. **Third Party Terms.** You acknowledge that software provided by third party vendors ("Third Party Software") may be embedded in or delivered with the Software. The terms of this Agreement and any other terms that Parasoft may specify will apply to such Third Party Software, and the Third Party Software vendors will be deemed third party beneficiaries under this Agreement. You may only use the Third Party Software with the Software. You may not use the Third Party Software on a stand-alone basis or use or integrate it with any other software or device.
- 2.7. **Evaluation License.** This Section 2.7 applies if Parasoft has provided the Software to You for evaluation purposes. Parasoft grants to You a thirty (30) day, limited license solely for the purpose of internal evaluation. You are strictly prohibited from using the Software for any production purpose or any purpose other than the sole purpose of determining whether to purchase a commercial license for the Software that You are evaluating. Parasoft is not

obligated to provide maintenance or support for the evaluation Software. YOU ACKNOWLEDGE THAT SOFTWARE PROVIDED FOR EVALUATION MAY (A) HAVE LIMITED FEATURES; (B) FUNCTION FOR A LIMITED PERIOD OF TIME; OR (C) HAVE OTHER LIMITATIONS NOT CONTAINED IN A COMMERCIAL VERSION OF THE SOFTWARE. NOTWITHSTANDING ANYTHING TO THE CONTRARY IN THIS AGREEMENT, PARASOFT IS PROVIDING THE EVALUATION SOFTWARE TO YOU "AS IS", AND PARASOFT DISCLAIMS ANY AND ALL WARRANTIES (INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND STATUTORY WARRANTIES OF NON-INFRINGEMENT), LIABILITIES, AND INDEMNIFICATION OBLIGATIONS OF ANY KIND. In the event of any conflict between this Section 2.7 and any other provision of this Agreement, this Section 2.7 will prevail and supersede such other provision with respect to Software licensed to You for evaluation purposes.

- 2.8. **Education License.** If You are an educational or academic institution and are receiving a discount from Parasoft, You may use the Software solely for education or academic purposes and You may not use the Software for any commercial purpose. Parasoft may require that You provide proof of your status as an educational or academic institution.
- 2.9. **Audit.** You shall maintain accurate business records relating to its use and deployment of the Software. Parasoft shall have the right, not more than once every twelve (12) months and upon ten (10) business days prior written notice, to verify Your compliance with its obligations under this Agreement by auditing Your business records and Your use and deployment of the Software within Your information technology systems. Parasoft and/or a public accounting firm selected by Parasoft shall perform the audit during Your regular business hours and comply with Your reasonable safety and security policies and procedures. Any agreement You may require the public accounting firm to execute shall not prevent disclosure of the audit results to Parasoft. You shall reasonably cooperate and assist with such audit. You shall, upon demand, pay to Parasoft all license and Maintenance fees for any unauthorized deployments and/or excess usage of Software products disclosed by the audit. License fees for such unauthorized deployments and/or excess usage shall be invoiced to and paid by You at Parasoft's then-current list price, and applicable Maintenance fees shall be applied retroactively to the entire period of the unauthorized and/or excess usage. Parasoft shall be responsible for its own costs and expenses in conducting the audit, unless the audit indicates that You have exceeded its Licensed Capacity or otherwise exceeds its license restrictions, such that the then-current list price of non-compliant Software deployment exceeds five percent (5%) of the total then-current list price of the Software actually licensed by You, in which event You shall, upon demand, reimburse Parasoft for all reasonable costs and expenses of the audit.
3. **TITLE.** Parasoft retains all right, title and interest in and to the Software and User Documentation and all copies, improvements, enhancements, modifications and derivative works of the Software and User Documentation, including, without limitation, all patent, copyright, trade secret, trademarks and other intellectual property rights. You agrees that it shall not, and shall not authorize others to, copy (except as expressly permitted herein), make modifications to, translate, disassemble, decompile, reverse engineer, otherwise decode or alter, or create derivative works based on the Software or User Documentation. Except as otherwise provided, Parasoft grants no express or implied rights under this license to any of Parasoft's patents, copyrights, trade secrets, trademarks, or other intellectual property rights.

#### 4. TERMINATION

- 4.1. **Default; Bankruptcy.** Parasoft may terminate this Agreement if (a) You fail to pay any amount when due under any order You have placed with Parasoft and do not cure such non-payment

within ten (10) days of receipt of written notice of non-payment; (b) You materially breach this Agreement and do not cure such breach within thirty (30) days of receipt of written notice of such breach; (c) subject to provisions of applicable bankruptcy and insolvency laws, You become the subject of any involuntary proceeding relating to insolvency and such petition or proceeding is not dismissed within sixty (60) days of filing; or (d) You become the subject of any voluntary or involuntary petition pursuant to applicable bankruptcy or insolvency laws, or request for receivership, liquidation, or composition for the benefit of creditors and such petition, request or proceeding is not dismissed within sixty (60) days of filing.

- 4.2. **Effect of Termination.** Upon termination of this Agreement, You shall immediately discontinue use of, and uninstall and destroy all copies of, all Software. Within ten (10) days following termination, You shall certify to Parasoft in a writing signed by an officer of Yours that all Software has been uninstalled from Your computer systems and destroyed.

## 5. LIMITED WARRANTY

- 5.1. **Performance Warranty.** Parasoft warrants that the Software, as delivered by Parasoft and when used in accordance with the User Documentation and the terms of this Agreement, will substantially perform in accordance with the User Documentation for a period of ninety (90) days from the date of initial delivery of the Software. If the Software does not operate as warranted and You have provided written notice of the non-conformity to Parasoft within the ninety (90) day warranty period, Parasoft shall at its option (a) repair the Software; (b) replace the Software with software of substantially the same functionality; or (c) terminate the license for the nonconforming Software and refund the applicable license fees received by Parasoft for the nonconforming Software. The foregoing warranty specifically excludes defects in or non-conformance of the Software resulting from (a) use of the Software in a manner not in accordance with the User Documentation; (b) modifications or enhancements to the Software made by or on behalf of You; (c) combining the Software with products, software, or devices not provided by Parasoft; or (d) computer hardware malfunctions, unauthorized repair, accident, or abuse.
- 5.2. **Disclaimers.** THE WARRANTIES SET FORTH IN THIS SECTION 5 ARE EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, AND PARASOFT EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND STATUTORY WARRANTIES OF NON-INFRINGEMENT. PARASOFT DOES NOT WARRANT THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS OR THAT USE OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE. THE REMEDIES SET FORTH IN THIS SECTION 5 ARE YOUR SOLE AND EXCLUSIVE REMEDIES AND PARASOFT'S SOLE AND EXCLUSIVE LIABILITY REGARDING FAILURE OF ANY SOFTWARE TO FUNCTION OR PERFORM AS WARRANTED IN THIS SECTION 5.

## 6. INDEMNIFICATION

- 6.1. **Infringement.** Parasoft shall defend any claim against You that the Software infringes any intellectual property right of a third party, provided that the third party is located in a country that is a signatory to the Berne Convention, and shall indemnify You against any and all damages finally awarded against You by a court of final appeal, or agreed to in settlement by Parasoft and attributable to such claim, so long as You (a) provide Parasoft prompt written notice of the claim; (b) provide Parasoft all reasonable assistance and information to enable Parasoft to perform its duties under this Section 6; (c) allow Parasoft sole control of the defense and all related settlement negotiations; and (d) have not compromised or settled such claim. If the Software is found to infringe, or if Parasoft determines in its sole opinion that it is likely to be found to infringe, then Parasoft may, at its option (a) obtain for You the right to continue to use the Software; (b) modify the Software to be non-infringing or replace it with a non-infringing functional equivalent, in which case You shall stop using any infringing version

of the Software; or (c) terminate Your rights and Parasoft's obligations under this Agreement with respect to such Software and refund to You the unamortized portion of the Software license fee paid for the Software based on a five year straight-line depreciation schedule commencing on the date of delivery of the Software. The foregoing indemnity will not apply to any infringement resulting from (a) use of the Software in a manner not in accordance with the User Documentation; (b) modifications or enhancements to the Software made by or on behalf of You; (c) combination, use, or operation of the Software with products not provided by Parasoft; or (d) use of an allegedly infringing version of the Software if the alleged infringement could be avoided by the use of a different version of the Software made available to You.

6.2. **Disclaimers.** THIS SECTION 6 STATES YOUR SOLE AND EXCLUSIVE REMEDY AND PARASOFT'S SOLE AND EXCLUSIVE LIABILITY REGARDING INFRINGEMENT OR MISAPPROPRIATION OF ANY INTELLECTUAL PROPERTY RIGHTS OF A THIRD PARTY.

7. **LIMITATION OF LIABILITY.** IN NO EVENT SHALL PARASOFT OR ITS THIRD PARTY VENDORS BE LIABLE TO YOU OR ANY OTHER PARTY FOR (A) ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR (B) LOSS OF DATA, LOSS OF PROFITS, BUSINESS INTERRUPTION, OR SIMILAR DAMAGES OR LOSS, EVEN IF PARASOFT AND ITS THIRD PARTY VENDORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT AS LIMITED BY APPLICABLE LAW AND EXCLUDING PARASOFT'S LIABILITY TO YOU UNDER SECTION 6 (INDEMNIFICATION), AND REGARDLESS OF THE BASIS FOR YOUR CLAIM, PARASOFT'S MAXIMUM LIABILITY UNDER THIS AGREEMENT WILL BE LIMITED TO THE LICENSE OR MAINTENANCE FEES PAID FOR THE SOFTWARE OR MAINTENANCE GIVING RISE TO THE CLAIM. THE FOREGOING LIMITATIONS WILL APPLY NOTWITHSTANDING THE FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY

8. **CONFIDENTIAL INFORMATION.** For purposes of this Agreement, "Confidential Information" will include trade secrets contained within the Software and User Documentation, the terms and pricing of the Software and Maintenance (including any pricing proposals), and such other information (a) identified by either party as confidential at the time of disclosure or (b) that a reasonable person would consider confidential due to its nature and circumstances of disclosure ("Confidential Information"). Confidential Information will not include information that (a) is or becomes a part of the public domain through no act or omission of the receiving party; (b) was in the receiving party's lawful possession prior to receiving it from the disclosing party; (c) is lawfully disclosed to the receiving party by a third party without restriction on disclosure; or (d) is independently developed by the receiving party without breaching this Agreement. Each party agrees to maintain all Confidential Information in confidence and not disclose any Confidential Information to a third party or use the Confidential Information except as permitted under this Agreement. Each party shall take all reasonable precautions necessary to ensure that the Confidential Information is not disclosed by such party or its employees, agents or authorized users to any third party. Each party agrees to immediately notify the other party of any unauthorized access to or disclosure of the Confidential Information. The receiving party agrees that any breach of this Section 8 may cause irreparable harm to the disclosing party, and such disclosing party shall be entitled to seek equitable relief in addition to all other remedies provided by this Agreement or available at law.

## 9. MAINTENANCE

9.1. **Maintenance Period.** You are required to purchase first year Maintenance with the Software, and the Maintenance period will commence upon the initial delivery of the Software and continue for a period of one year. The Maintenance period, at Your option, may be renewed

pursuant to subsequent orders. Prior to such renewal, Parasoft may, upon ten (10) business days written notice, require You to provide a report on Your use and deployment of the Software. Such report will be certified by an officer of Yours and will specify, with respect to Your Software: (a) the type and amount of Licensed Capacity; (b) the version; and (c) the Parasoft license serial number. Parasoft shall issue an annual renewal notice to You at least ninety (90) days prior to the expiration of the then-current Maintenance period. Maintenance fees will be based on the then-current list price and are subject to change without notice.

- 9.2. **Support Coordinators.** Maintenance will consist of support services provided by Parasoft to one designated support coordinator of Yours (and one backup coordinator) per Your location, by telephone, email, and website. Support is available during normal business hours in the applicable location within the Territory, Monday through Friday, excluding nationally observed holidays.
- 9.3. **Additional Licensed Capacity.** In the event that You purchases additional Licensed Capacity for the Software prior to the annual anniversary date of the Maintenance period, You agree to pay applicable Maintenance fees based on Parasoft's then-current Maintenance rates. Maintenance fees will apply from the effective date of such additional Licensed Capacity and continue for a period of one year thereafter, unless otherwise agreed to in writing by the parties, so that Maintenance for Your previously acquired Software and added Licensed Capacity is coterminous.
- 9.4. **New Releases.** During any period in which You are current on Maintenance, Parasoft shall provide You with any new release of the Software, which may include generally available error corrections, modifications, maintenance patch releases, enhancements (unless priced separately by Parasoft and generally not included with new licenses for the Software at that time), and the revised User Documentation, if applicable. Notwithstanding the foregoing, stand-alone error corrections that are not part of a new release will not be independently supported but will be incorporated into the next release of the Software. If You install a new release of the Software, You may continue to use the previous version of the Software for up to ninety (90) days in order to assist You in the transition to the new release. Once You complete its transition to the new release of the Software, You must discontinue use of the previous version of the Software.
- 9.5. **Supported Releases.** Parasoft shall continue to support the immediately preceding release of the Software for a period of twelve (12) months following the discontinuance of such Software or the date on which the new release becomes generally available, provided that You have paid applicable Maintenance fees and incorporated all Maintenance patch releases issued by Parasoft for the release of the Software.
- 9.6. **Reinstatement of Maintenance.** If You allow Maintenance to expire, You may, at a later date, renew Maintenance by paying the following: (a) if You have installed the current release of the Software but have failed to pay the applicable renewal fee on or before the ninetieth (90th) day following expiration of the Maintenance period, annual Maintenance fees at Parasoft's then-current rates, plus Parasoft's then-current reinstatement fee; or (b) if You have not installed the current release of the Software or have failed to pay the applicable renewal fee by the ninetieth (90th) day following expiration of the Maintenance period, annual Maintenance fees at Parasoft's then-current rates, plus Parasoft's then-current license update fee for the current release of the Software.

## 10. GENERAL

- 10.1. **Independent Contractors.** The parties acknowledge and agree that each is an independent contractor. This Agreement will not be construed to create a partnership, joint venture or agency relationship between the parties.

- 10.2.**Entire Agreement.** The terms and conditions of this Agreement apply to all Software licensed, all User Documentation provided, and all Maintenance purchased hereunder. This Agreement will supersede any different, inconsistent or preprinted terms and conditions in any order form of Yours, purchase order or other ordering document.
- 10.3.**Assignment.** You have no right to assign, sublicense, pledge, or otherwise transfer any of Your rights in and to the Software, User Documentation or this Agreement, in whole or in part (collectively, an "Assignment"), without Parasoft's prior written consent, and any Assignment without such consent shall be null and void. Any change in control of Your organization or entity, whether by merger, share purchase, asset sale, or otherwise, will be deemed an Assignment subject to the terms of this Section 13.3.
- 10.4.**Force Majeure.** No failure, delay or default in performance of any obligation of a party to this Agreement, except payment of license fees due hereunder, will constitute an event of default or breach of the Agreement to the extent that such failure to perform, delay or default arises out of a cause, existing or future, that is beyond the reasonable control of such party, including, without limitation, action or inaction of a governmental agency, civil or military authority, fire, strike, lockout or other labor dispute, inability to obtain labor or materials on time, flood, war, riot, theft, earthquake or other natural disaster ("Force Majeure Event"). The party affected by such Force Majeure Event shall take all reasonable actions to minimize the consequences of any Force Majeure Event.
- 10.5.**Severability.** If any provision of this Agreement is held to be illegal or otherwise unenforceable by a court of competent jurisdiction, that provision will be severed and the remainder of the Agreement will remain in full force and effect.
- 10.6.**Waiver.** The waiver of any right or election of any remedy in one instance will not affect any rights or remedies in another instance. A waiver will be effective only if made in writing and signed by an authorized representative of the applicable party.
- 10.7.**Notices.** All notices required by this Agreement will be in writing, addressed to the party to be notified and deemed to have been effectively given and received (a) on the fifth business day following deposit in the mail, if sent by first class mail, postage prepaid; (b) upon receipt, if sent by registered or certified U.S. mail, postage prepaid, with return receipt requested; (c) upon transmission, if sent by facsimile and confirmation of transmission is produced by the sending machine and a copy of such facsimile is promptly sent by another means specified in this Section 10.7; or (d) upon delivery, if delivered personally or sent by express courier service and receipt is confirmed by the recipient. Notices will be addressed to the parties based on the address stated in the applicable order, to the attention of the Legal Department. A change of address for notice purposes may be made pursuant to the procedures set forth above.
- 10.8.**Export Restrictions.** You acknowledge that the Software and certain Confidential Information (collectively "Technical Data") are subject to United States export controls under the U. S. Export Administration Act, including the Export Administration Regulations, 15 C.F.R. Parts 730 et seq. (collectively, "Export Control Laws"). Each party agrees to comply with all requirements of the Export Control Laws with respect to the Technical Data. Without limiting the foregoing, You shall not (a) export, re-export, divert or transfer any such Technical Data, or any direct product thereof, to any destination, company, or person restricted or prohibited by Export Control Laws; (b) disclose any such Technical Data to any national of any country when such disclosure is restricted or prohibited by the Export Control Laws; or (c) export or re-export the Technical Data, directly or indirectly, for nuclear, missile, or chemical/biological weaponry end uses prohibited by the Export Control Laws.
- 10.9.**U. S. Government Rights.** The Software and User Documentation are deemed to be "commercial computer software" and "commercial computer software documentation" as defined in FAR Section 12.212 and DFARS Section 227.7202, as applicable. Any use, modification, reproduction, release, performance, display, or disclosure of the Software and

User Documentation by the United States government will be solely in accordance with the terms of this Agreement

10.10.**Choice of Law; Jurisdiction.** This Agreement is governed by and construed in accordance with the laws of the State of California, U. S. A., exclusive of any provisions of the United Nations Convention on Contracts for the International Sale of Goods, including any amendments thereto, and without regard to principles of conflicts of law. Any suits concerning this Agreement will be brought in the federal courts for the Central District of California or the state courts in Los Angeles County, California. The parties expressly agree that the Uniform Computer Information Transactions Act, as adopted or amended from time to time, shall not apply to this Agreement or the Software and Maintenance provided hereunder.

10.11.**Amendment.** This Agreement may only be modified by a written document signed by an authorized representative of Parasoft and by You.

10.12.**Survival.** Any terms of this Agreement which by their nature extend beyond the termination or expiration of this Agreement will remain in effect. Such terms will include, without limitation, all provisions herein relating to limitation of liability, title and ownership of Software, and all general provisions.

Copyright 1993-2017  
Parasoft Corporation  
101 East Huntington Drive, 2nd Floor  
Monrovia, CA 91016 USA

# PARASOFT 使用許諾契約書

本和文契約書は、英文契約書を和訳したものです。万一和英契約書の表現に不一致があった場合は英文契約書に従うこととなりますことをご了承下さい。可能な限り忠実なる対訳に努めました但上記ご了解をいただくために必要な方は、英文契約書を必ず慎重にお読みください。

本使用許諾契約（「本契約」）をソフトウェアを使用する前に注意深くお読みください。Parasoft 社（「Parasoft」）はお客様が本契約のすべての条項に同意した場合のみ、ソフトウェアを使用する個人または企業（「お客様」あるいは「お客様の」）に対して本ソフトウェアの使用を許諾します。この契約はお客様と Parasoft との間を法的に拘束する契約です。[同意します] あるいは [はい] ボタンをクリックするかその他の方法で電子的に同意を示すこと、もしくは本ソフトウェアをインストールすることにより、お客様は本契約の諸条件に同意したものと見なされ、加えてお客様は本契約がお客様自らが書面をもって同意し署名した契約と同様の拘束力があるものであることに同意したものと見なされます。お客様がこれらの諸条件に同意されない場合は、[同意しません] あるいは [いいえ] ボタンをクリックし、本ソフトウェアをそれ以上使用しないでください。

## 1. 用語の定義

- 1.1. 「同時利用ユーザー」は、任意の時点で、直接あるいはアプリケーションを経由して本ソフトウェアに接続した人を意味します。
- 1.2. 「インスタンス」は、1 台のマシン上でソフトウェアの初期化あるいは実行の単一の発生を意味します。
- 1.3. 「ライセンス利用範囲」は、注文書に明記された料金設定による、同時利用ユーザーベース、ノードロックマシンベース、インスタンスベースを含みますがそれらに限らない最大利用数および / あるいは利用範囲を指定するライセンスを意味します。
- 1.4. 「保守サービス」は、発注書に明記され、かつ、本契約に従って Parasoft が提供する、本ソフトウェアの保守ならびにテクニカルサポートサービスを意味します。
- 1.5. 「ノードロック」は、許諾された本ソフトウェアの単一インスタンスを実行させる単一のマシンのためのライセンスを意味します。ノードロックライセンスはユーザーが実際に存在することを要求します。また、リモート環境でマシンに接続したり本ソフトウェアを使用したりすることはできません。
- 1.6. 「本ソフトウェア」は、お客様が発注した時点で販売できる状態にあり、かつ、発注書類に明記されたオブジェクトコード形式の Parasoft のソフトウェア製品および保守サービスの一環として Parasoft により提供された変更、修正、アップデートを意味します。
- 1.7. 「許諾地域」は、お客様が本ソフトウェアの発注書に本ソフトウェアを使用する権利を有する国として記載した国、もしくは、発注時に指定されていない場合はお客様の発注書が発行された国を意味します。
- 1.8. 「ユーザードキュメント」は本ソフトウェアに関連した、ユーザーズガイド、インストールガイド、および / あるいは、オンラインドキュメントを意味します。「ユーザードキュメント」には、販促関連資料、あるいは、提案依頼に対する回答は含まれません。

## 2. ライセンスの許諾および本ソフトウェアの使用

### 2.1. ライセンス許諾

本契約の諸条件に従って、Parasoft はお客様に本ソフトウェアを許諾地域内で、ユーザードキュメントのとおり、許諾されたライセンス利用範囲に則って利用する永久的なあるいは年間契約（注文書に明記されている）の非独占ライセンスを提供します。お客様は、本契約が Parasoft によってここに規定されている本ソフトウェアのライセンスを許諾する唯一のものであり、本ソフトウェアの販売を可能にするものではないことを認め、同意します。お

お客様は、契約によっても著作権保護法の規定によっても、本ソフトウェアを再販する権利を有することはありません。

## 2.2. 使用権

お客様は、本ソフトウェアおよび/またはユーザードキュメントをお客様の内部事業運営のため、およびお客様のデータを処理するためだけに使用することができます。お客様は、次の(a)から(e)を行うことはできません。(a) 第三者あるいはライセンスを持たない者に本ソフトウェアあるいはユーザードキュメントを使用することを許可すること、(b) お客様あるいはお客様の関係会社のデータではないデータに対して使用すること、もしくは、使用することを許可すること、(c) 本ソフトウェアをサービスの作業で使用すること、(d) 本ソフトウェアあるいはユーザードキュメントを第三者にサブライセンス、レンタルあるいはリースすること、(e) 本ソフトウェアあるいはユーザードキュメントに係わるベンチマークあるいはその他の比較を実行する、公表する、もしくは第三者に公開すること。お客様は、Parasoft から最初に購入したライセンスの数量以外に、本ソフトウェアを複数の、または分割された、もしくは仮想化されたコンピュータで同時に使用することはできません。お客様は、Parasoft が事前に書面で同意した場合を除き、第三者のアウトソーシング業者にお客様に代わって本ソフトウェアを使用することを許可することはできません。前述にかかわらず、お客様はお客様の関係会社ならびに第三者の契約会社に対して本ソフトウェアをお客様のためにのみ使用することを許可することができます。ただし、お客様は当該関係会社および契約会社が本契約を遵守することに全責任を負います。

## 2.3. ライセンスキー

お客様は、本ソフトウェアには本ソフトウェアの機能性、ならびに本ソフトウェアとともに配布される、もしくは本ソフトウェアに組み込まれているサードパーティソフトウェアの機能性を有効にする1つあるいはそれ以上のライセンスキーが含まれることを了承します。お客様がParasoftより発行されたライセンスキーでのみ、本ソフトウェアに接続し使用することができます。お客様は、ライセンスキーを編集、改ざん、リバースエンジニアリング、逆コンパイル、もしくは逆アセンブリすることはできません。Parasoftから本ソフトウェアの新しいライセンスキーが発行された場合は、お客様は以前に発行された本ソフトウェアの実行を可能にするライセンスキーを使用することはできません。

## 2.4. 保存用複製

お客様は、本ソフトウェアのコピーを1つ、バックアップあるいは保存の目的でのみ作成することができます。お客様は、ユーザードキュメントについて妥当な数のコピーをお客様の内部使用のため作成することができます。本ソフトウェアおよびユーザードキュメントのすべての複製には、原本に表示されている著作権および類似する財産権によって保護されている旨の通知を含めなければなりません。本ソフトウェアに係わるすべての人が本契約で規定されているお客様の義務を負い、かつ、お客様がこれらの義務を遵守するため合理的な予防措置を講ずるならば、本ソフトウェアの複製を離れた場所に保管することができます。Parasoftは、Parasoftがお客様が本契約で規定されている義務の遵守を怠ったと信じる合理的な根拠が存在する場合、著作権ならびに財産権で保護されている製作物の複製を作成する許可を取り消す権利を有します。

## 2.5. ライセンス利用範囲

Parasoftは、本ソフトウェアのライセンスをライセンス利用範囲に基づいて同時利用ユーザーあるいはノードロックマシンを含むがこれらに限らないさまざまな利用形態で提供します。同時利用ユーザーライセンスは、本ソフトウェアに接続する同時利用ユーザー数が購入した同時利用ユーザー数の合計を上回らない限り、複数の同時利用ユーザーが本ソフトウェアの接続を共有し使用することができます。ノードロックライセンスは、単独の特定のマシンで本ソフトウェアの単一インスタンスを実行することができます。本ソフトウェアが接続するアプリケーションがそのアプリケーションのユーザーに本ソフトウェア、もしくは本ソフトウェアによって処理されたデータへの接続を許可する、多重処理方式、データベースもしくはWebポータルアプリケーションであった場合、各アプリケーションに対して別個の

同時利用ユーザーライセンスが要求されます。利用形態に係わらず、お客様は、速やかに Parasoft にライセンス利用範囲を超えた利用の増加を書面で通知するものとします。お客様は、ライセンス利用範囲に係わるライセンスの増加分のライセンスを取得しなければなりません。また、お客様は、本ソフトウェアの追加ライセンスに対する費用を Parasoft のその時点の最新の価格表に則って Parasoft に支払うことに同意するものとします。

## 2.6. 第三者の条件

お客様は、サードパーティベンダー（「サードパーティのソフトウェア」）によって提供されるソフトウェアが本ソフトウェアとともに提供されること、もしくは本ソフトウェアに組み込まれていることを了承します。本契約の諸条件と Parasoft が特定するその他の条件は、当該サードパーティのソフトウェアに適用され、サードパーティのソフトウェアのベンダーは本契約に基づいて第三受益者と見なされます。お客様は、サードパーティのソフトウェアを本ソフトウェアとともにのみ使用することができます。お客様は、サードパーティのソフトウェアを単独で使用する、またはその他のソフトウェアもしくはデバイスと結合あるいは使用することはできません。

## 2.7. 評価用ライセンス

この 2.7 章は、Parasoft が本ソフトウェアをお客様に評価目的で使用するものとして提供した場合に適用されます。Parasoft は、お客様にお客様の組織内での評価を目的とするためだけに 30 日間の制限付きライセンスを提供します。お客様は、本ソフトウェアを製造目的で使用すること、もしくはお客様が評価している本ソフトウェアの商用ライセンスを購入するかどうかの判断のため以外の目的で使用することを厳密に禁止されています。Parasoft は、本ソフトウェアの評価使用に対する保守サービスあるいはサポートを提供する義務はありません。お客様は、評価用の本ソフトウェアは、(A) 機能に制限があること、(B) 時間的な制限があること、(C) 本ソフトウェアの商用バージョンには含まれないその他の制限があることを了承します。本契約の他の規定に係わらず、Parasoft は、評価用の本ソフトウェアをお客様に、「現状のまま」で提供し、Parasoft は、いかなる保証（商品性、特定目的への適合性、第三者の権利を侵害していないことに対する保証を含みますがこれに限りません。）、責任、その他のあらゆる賠償義務を負いません。この 2.7 章と本契約の他の条項との間に不一致がある場合は、評価目的のためお客様に提供された本ソフトウェアについては、この 2.7 章が優先され、かかるその他の条項は本条項に置き換えられます。

## 2.8. 教育機関向けライセンス

お客様が教育あるいは学術研究機関である場合、Parasoft よりディスカウントを受けることができます。お客様は本ソフトウェアを教育あるいは学術研究のみを目的として使用することができますが、お客様は本ソフトウェアを営利を目的として使用することはできません。Parasoft は、お客様が教育あるいは学術研究機関であることの証明を要求することができます。

## 2.9. 監査

お客様は、本ソフトウェアの使用および配布に係わる取引記録を正確に保管するものとします。Parasoft は、12 か月につき 1 回以下の割合で 10 営業日前の書面による通知をもって、お客様の本ソフトウェアの取引記録ならびに情報処理システム内の本ソフトウェアの使用および配布について監査することにより、お客様の本契約における義務の遵守を検証する権利を有するものとします。Parasoft および / または Parasoft によって指定された公認会計士事務所は、お客様が通常の営業時間に監査を実施するものとし、お客様の適切な安全および保安ポリシーならびに手順を遵守するものとします。お客様が公認会計士事務所を実施を要求できるいかなる契約があったとしても、Parasoft への監査結果の報告を拒否することは認められません。お客様は、当該監査に協力・支援するものとします。お客様は、必要に応じて、Parasoft に監査によって明らかになった許諾されていない配布および / または超過した使用に利用された本ソフトウェア製品についてすべてのライセンスと保守サービスの費用を支払うものとします。許諾されていない配布および / または超過した使用に該当するライセ

ンスの費用は請求され、お客様により Parasoft のその時点の最新の価格で支払われるもの  
とします。また、適用される保守サービス費用は、許諾されていない配布および/または超  
過した使用の全期間について、さかのぼって適用されます。Parasoft は、監査により、お客  
様がライセンス利用範囲を超えて利用していたことが確認された、もしくはライセンスの制  
限数を超過していたことが明らかになった場合を除き、監査の実施に伴う費用と支出を負担  
します。お客様が許諾されていない本ソフトウェアを配布したり、超過して使用していた場  
合、許諾されていない本ソフトウェアの配布についてその時点の最新の価格表に則って算出  
した金額が、実際にお客様に許諾されているすべての本ソフトウェアについてその時点の最  
新の価格表に則って計算した合計金額の 5% を超えていた場合、お客様は、必要に応じて、  
監査にかかった費用と支出の全額を Parasoft に返済するものとします。

### 3. 権原

Parasoft は、本ソフトウェアおよびユーザードキュメントについて、複製、改善、機能追  
加、変更、本ソフトウェアおよびユーザードキュメントの派生物に係わる、すべての特許、  
著作権、企業秘密、登録商標、その他の知的財産権を含むがこれらに限らないすべての権利、  
権原、利益を有します。お客様は、本ソフトウェアあるいはユーザードキュメントについて、  
複製（本契約に明記されるものを除く）、変更の作成、翻訳、逆アセンブリ、逆コンパイル、  
リバースエンジニアリング、デコードあるいは改変、派生物の作成を実施することは認めら  
れていないことに同意します。また、お客様はこれらについて他者に実施の許可を与えるこ  
とも認められていないことに同意します。特別の定めがある場合を除き、Parasoft は、明示  
あるいは暗黙を問わず、Parasoft の特許、著作権、企業秘密、登録商標、その他知的財産権  
のいずれかに係わる権利を許諾しません。

### 4. 契約の解除

#### 4.1. 不履行；破産

Parasoft は、次の (a) から (d) のいずれかの状態になったとき、本契約を解除することがで  
きます。(a) お客様が Parasoft に発行した注文書に記載されている支払期日に全額が支払わ  
れず、書面による未払いの通知を受け取ってから 10 日以内に未払いが是正されない、(b)  
お客様が本契約に対して重大な違反をし、書面による違反の通知を受け取ってから 30 日以  
内にかかる違反が是正されない、(c) 適用可能な破産法あるいは倒産法の条項に基づき、お  
客様が倒産に伴う強制的手続きの対象になり、該当する申し立てあるいは手続きが 60 日以  
内に終了しない、(d) お客様が適用可能な破産法あるいは倒産法に基づき自発的あるいは強  
制的な申し立ての対象になる、もしくは債権者のため倒産処理手続、清算手続、債務減免手  
続を要求する場合で、かつ、該当する申し立て、要求、手続きが、60 日以内に終了しない。

#### 4.2. 解除の履行

本契約を解除された場合には、お客様は速やかに本ソフトウェアの使用を中止し、アンイン  
ストールし、すべての複製を破棄するものとします。解除から 10 日以内にお客様は  
Parasoft にすべての本ソフトウェアがお客様のコンピュータシステムからアンインストール  
され破棄されたことを証明する書類にお客様の役員が署名したものを送付するものとします。

### 5. 限定保証

#### 5.1. 動作保証

Parasoft は、本ソフトウェアが Parasoft から納品され、かつ、ユーザードキュメントと本  
契約の諸条件に従って使用された場合に、本ソフトウェアがユーザードキュメントの記載内  
容のとおり実質的に動作することを、本ソフトウェアの最初の納品日から 90 日間保証し  
ます。本ソフトウェアが、保証内容のように動作せず、お客様が 90 日間の保証期間の間に  
Parasoft に不具合を書面で通知した場合は、Parasoft は、任意の選択として、次の (a) か  
ら (c) を実施するものとします。(a) 本ソフトウェアを修正する、(b) 本ソフトウェアを実質  
的に同じ機能を持つソフトウェアに交換する、(c) 不具合のある本ソフトウェアのライセン

スを終了し、Parasoft が不具合のある本ソフトウェアの対価として受け取ったライセンスの費用を払い戻す。前述の保証は、明確に本ソフトウェアの欠陥あるいは不具合が次の (a) から (d) によるものの場合、保証の対象から外します。(a) ユーザードキュメントに反する本ソフトウェアの使用、(b) お客様による本ソフトウェアへの変更あるいは機能追加、(c) 本ソフトウェアと Parasoft によって提供されたものでない製品、ソフトウェアもしくはデバイスとの統合、(d) コンピュータハードウェアの故障、認められていない修理、事故、誤用のいずれか。

## 5.2. 免責事項

5 章に明記されている保証は、唯一の、すべての他の保証に代わるものであり、Parasoft は、明示あるいは暗黙を問わず、すべてのその他の一切の保証の責任を負いません。これには、商品性、特定目的への適合性、第三者の法的権利を侵害していないことを含みますがこれに限りません。Parasoft は、本ソフトウェアがお客様の要求を満たす、もしくは本ソフトウェアを途切れることなく使用できる、あるいはエラーのない状態になることを保証しません。5 章に明記されている救済措置は、5 章に関するお客様に対する唯一の救済措置であり、5 章で保証するすべてのソフトウェアの実行あるいは実施の失敗に係わる Parasoft の唯一の保証です。

## 6. 補償

### 6.1. 侵害

Parasoft は、第三者がベルヌ条約に加盟している国に所在するならば、本ソフトウェアが第三者の知的財産権を侵害しているというお客様に対する請求について防御するものとし、上告裁判所によって最終的にお客様に裁定が下った場合の損害、Parasoft によって和解が成立した当該請求に起因する損害について、お客様が次の (a) から (d) を守る限り、賠償するものとし、(a) Parasoft に速やかに書面で請求の通知を送付する、(b) Parasoft に Parasoft が 6 章に基づく義務を果たすために必要な合理的な支援と情報を提供する、(c) Parasoft に独自の弁護活動とすべての和解交渉活動の統括を許可する、(d) 和解または示談をしない。本ソフトウェアが侵害していることを発見された、もしくは、Parasoft が侵害している可能性があるとして独自に判断したならば、Parasoft は、任意の選択として、次の (a) から (c) の措置を講ずることができます。(a) お客様に本ソフトウェアの使用を継続する権利を提供する、(b) お客様が侵害しているバージョンの本ソフトウェアの使用を中止できるよう、本ソフトウェアを侵害していない状態に修正する、もしくは機能的に同等なもので侵害していないものに置き換える、(c) 該当する本ソフトウェアについて本契約で規定されているお客様の権利と Parasoft の義務を終了し、お客様に本ソフトウェアのライセンス購入額の未償却分として 5 年の減価償却定額法をもとに本ソフトウェアの納品日から起算した額を払い戻す。前述の賠償は、次の (a) から (d) の結果として発生した侵害には適用されません。(a) ユーザードキュメントに反する本ソフトウェアの使用、(b) 本ソフトウェアに対してお客様によって、あるいは、お客様のために施された変更あるいは機能追加、(c) Parasoft から提供されたものでない製品と本ソフトウェアの統合、使用、操作、(d) 侵害の申し立てが、本ソフトウェアのお客様が使用できる他のバージョンを使用することにより回避できるにも係わらず、侵害しているバージョンの本ソフトウェアを使用すること。

### 6.2. 免責事項

6 章は、第三者の知的財産権に対する侵害または不正目的使用に係わるお客様に対する唯一の救済措置および Parasoft の唯一の保証です。

## 7. 責任制限

Parasoft あるいはそのサードパーティベンダーは、お客様あるいはその他の当事者に、(A) いかなる特別、付随的、間接的、または結果的損失、(B) データの喪失、利益の損失、事業の中断、類似の経済的損失あるいは損害に対する一切の責任を負いません。これは、

Parasoft あるいはサードパーティベンダーがかかる損害が発生する可能性を予め警告されていたとしても変わりません。適用法と 6 章（補償）に基づく Parasoft のお客様への唯一の賠償責任を除いて、また、お客様の請求の基礎に関係なく、本契約に基づく Parasoft の最大賠償額は、請求の契機となった本ソフトウェアあるいは保守サービスの対価としてお客様が支払った本ソフトウェアあるいは保守サービスの額を超えることはありません。前述の制限は、救済措置の主たる目的を果たすことができるかどうかに関わらず、適用されます。

## 8. 機密情報

本契約の解釈上、機密情報には、本ソフトウェアおよびユーザードキュメントに含まれる企業秘密、本ソフトウェアおよび保守サービスの諸条件あるいは価格（提案価格を含む）、ならびに、次の (a) および (b) のようなその他の情報、(a) いずれかの当事者によって開示の時点で機密であると判断された情報、(b) 一般的にその性質や開示の状況から機密であると考えられる情報、が含まれます（「機密情報」）。機密情報には、次の (a) から (d) に該当する情報は含まれません。(a) 公知の情報、もしくは受領者の作為あるいは不作為による過失によって公知となった情報、(b) 受領者が開示者から得る以前から法的に正当に所有していた情報、(c) 開示の制限なしに第三者によって受領者に法的に正当に開示された情報、(d) 受領者によって本契約に違反することなく独自に開発された情報。両当事者は、本契約において認められている場合を除いて、すべての機密情報を機密に管理すること、ならびに機密情報を第三者に開示しないこと、あるいは機密情報を使用しないことを同意します。両当事者は、機密情報がかかる当事者あるいは従業員、もしくは、代理人あるいは正規ユーザーによって第三者に漏洩されないことを保証するのに必要なすべての合理的な予防措置を施すものとします。両当事者は、相手方に機密情報への不正なアクセス、あるいは、機密情報の漏洩の発生を速やかに通知することに同意します。受領者は、本 8 章に対する違反は開示者に回復することができない損害を与える可能性があることを了承します。かかる開示者は本契約あるいは適用法で定められたすべてのその他の救済措置に加えて法的救済を求める権利を有するものとします。

## 9. 保守サービス

### 9.1. 保守サービスの期間

お客様は、本ソフトウェアの購入と同時に最初の 1 年間の保守サービスを購入する必要があります。保守サービスの期間は、本ソフトウェアの最初の納品日から 1 年間です。保守サービス期間は、お客様の任意で更新することができます。Parasoft は、かかる更新の 10 営業日前の書面による通知で、お客様に本ソフトウェアに関するお客様の使用および配布に関するレポートの提出を要求することができます。かかるレポートは、お客様の役員によって承認されたものであり、かつ、お客様の本ソフトウェアについて、次の (a) から (c) に規定されている内容を明記するものとします。(a) 利用形態およびライセンス利用範囲の合計、(b) バージョン、(c) Parasoft ライセンス シリアル ナンバー。Parasoft あるいは Parasoft の正規販売代理店は、お客様に年間保守サービスの更新に係わる通知を遅くとも保守サービスの終了日の 90 日前までに発行するものとします。保守サービス費用は、その時点での最新の価格表に則って算出されます。また、保守サービス費用は、通知なしに変更される場合があります。

### 9.2. サポート担当者

保守サービスは、Parasoft あるいは Parasoft の正規販売代理店によるサポートサービスで、お客様の所在地につき、1 名の指定サポート担当者（加えて 1 名の副サポート担当者）に対して、電話、電子メール、Web サイトのいずれかで提供されます。サポート対応時間は、ライセンス利用範囲で指定された地域の祝日を除く、月曜日から金曜日、通常の営業時間です。

### 9.3. 追加のライセンス利用範囲

お客様が以前に導入した本ソフトウェアのライセンス利用範囲の年間保守サービスが終了日を迎える前にライセンス利用範囲を追加購入する場合、お客様は追加購入分の保守サービス

の費用について Parasoft のその時点での最新の保守サービス率に則って支払うことに同意します。保守サービス費用は、お客様が先に購入した本ソフトウェアの保守サービスと追加購入のライセンス利用範囲の保守サービスの終了日を一致させるよう両当事者が書面で同意しない限り、追加購入のライセンス利用範囲の有効日から 1 年間で適用されます。

#### 9.4. 新リリース

お客様が保守サービスに加入している間、Parasoft は、その時点で可能であれば、一般的に対応可能なエラーの修正、変更、パッチ、機能追加 (Parasoft によって別料金が課せられるもの、あるいは、その時点で本ソフトウェアに正式に含まれないものを除く)、ユーザードキュメントの改訂版を含む本ソフトウェアのあらゆる新リリース版を提供するものとします。前述にかかわらず、新リリース版に含まれないエラーの修正は、個別にサポートされることはありませんが、本ソフトウェアの次のリリース版に組み込まれる可能性があります。お客様は、本ソフトウェアの新リリース版をインストールした場合、新リリース版への移行をスムーズに行えるよう本ソフトウェアの前バージョンを 90 日間引き続き使用することができます。お客様が本ソフトウェアの新リリース版への移行を完了した場合には、お客様は本ソフトウェアの前バージョンの使用を終了しなければなりません。

#### 9.5. サポート対象リリース

Parasoft は、お客様が該当する保守サービス費用を支払っていて、かつ、Parasoft によって提供されたすべてのパッチを本ソフトウェアに適用しているならば、本ソフトウェアの直前のリリース版 (そのバージョンはバージョン番号の最初的小数点の右にある数字によって示される) のサポートを、本ソフトウェアの廃止から 12 か月間、もしくは、新リリース版が正式なものとなった日から 12 か月間、継続するものとします。

#### 9.6. 保守サービスの再加入

お客様が保守サービスを更新しなかった場合、お客様は、後日、次の (a) と (b) のいずれかを支払うことによって、保守サービスに再加入することができます。(a) お客様は本ソフトウェアの最新リリース版をインストールしているが、該当する更新費用を保守サービスの終了日から 90 日以内に支払わなかった場合は、Parasoft のその時点での最新の保守サービス率に則った年間保守サービス費用、加えて、Parasoft のその時点での最新の再加入費用、(b) お客様が本ソフトウェアの最新リリース版をインストールしていない場合、もしくは、該当する更新費用を保守サービス期間が終了してから 90 日を過ぎても支払わなかった場合、Parasoft のその時点での最新の保守サービス率に則った年間保守サービス費用、加えて、本ソフトウェアを最新リリース版にするための Parasoft のその時点での最新のアップデート費用。

### 10. 一般条項

#### 10.1. 当事者間の独立

両当事者は、互いに独立の契約者であることを認め、同意します。本契約は、両当事者の間にパートナーシップあるいは合併事業もしくは代理関係を創設すると解釈されることはありません。

#### 10.2. 完全合意

本契約の諸条件は、本契約に基づいて購入した、すべての本ソフトウェアのライセンス、すべてのユーザードキュメント、すべての保守サービスに適用されます。お客様の注文書、発注書、もしくはその他注文に係わる書類に本契約と一致しないあるいは矛盾した、もしくは予め印字された諸条件については、本契約が優先され、取って代わるものとします。

#### 10.3. 譲渡

お客様には、Parasoft が事前の書面で同意した場合を除き、本ソフトウェア、ユーザードキュメント、本契約に対する、あるいは、本契約に含まれるお客様の権利をすべてあるいは一部でも譲渡する、サブライセンスする、質権にする、移転する権利はありません (「譲渡」

)。かかる同意を得ていない譲渡は、無効であるものとします。合併、買収、売却、あるいは他の原因により、お客様の組織あるいは企業の支配権が移転した場合でも、10.3章にある譲渡と見なされます。

#### 10.4. 不可抗力

いずれかの当事者が本契約に基づくライセンス費用の支払いを除く本契約による義務の履行の失敗、遅延、不履行を犯したとしても、本契約による義務の不履行あるいは違反が存在するあるいは将来生ずるかかかる当事者の合理的な制御を超える事象に起因する、履行の失敗、遅延、不履行であるならば、かかる当事者に責任はないものとします。その事象とは、政府機関、市民・軍事機関による作為あるいは不作為、火災、ストライキ、ロックアウト、その他の労働争議、労働者あるいは材料の受け入れ困難、洪水、戦争、暴動、窃盗、地震、自然災害（「不可抗力」）を含みますがこれらに限りません。不可抗力の影響を受けた当事者は不可抗力の影響を最小限に留めるため、すべての合理的な措置を施すものとします。

#### 10.5. 可分性

本契約の条項が法律に反している、もしくは、管轄裁判所によって強制不能であると判断された場合でも、その条項を切り離れた本契約の残りの条項は完全効力が付与されるものとします。

#### 10.6. 権利の放棄

1つの事案についての権利あるいは救済措置の公示の放棄は、他の事案の権利あるいは救済措置には影響しません。放棄は書面に記載され、当該当事者の役員によって署名されたもののみが有効と見なされます。

#### 10.7. 通知

本契約に基づくすべての通知は書面でなされるものとし、次の (a) から (d) のいずれかをもって、通知される当事者の授受が有効だと思われる送付先に送付されたものと見なされます。(a) 第1種郵便かつ郵便料金前払いで投函された場合は投函から5営業日、(b) アメリカ合衆国郵便公社の書留郵便あるいは書留配達便で、かつ、郵便料金前払いの配達証明付きで配達された場合は受け取り、(c) ファクシミリで送信され、送信確認装置が搭載されているファクシミリ機から送信された、もしくは、かかるファクシミリのコピーが10.7章に規定された他の方法で即座に送付された場合は送信、(d) 人により配達され、もしくは、速配サービスで送付され、受け取りが受取人によって確認された場合は受け取り。いずれの当事者の正式な注文に記載されている住所あるいは法務部門あてに通知が送付されるものとします。通知先住所の変更は上記の手続きにより実施することができます。

#### 10.8. 輸出制限

お客様は、本ソフトウェアと一定の機密情報（「技術データ」）は輸出管理規則（15 CFR Part 730 et. seq.）を含む米国輸出管理法（「輸出管理法」）に基づく米国輸出規制の対象であることを認識します。両当事者は技術データについて輸出管理法のすべての規制に従うことに同意します。前述に限定することなく、お客様は次の (a) から (c) を実施することはできません。(a) 技術データあるいはその直近の製品を輸出管理法で禁止あるいは規制されている地域、企業、個人に対して輸出、再輸出、移転、譲渡すること、(b) 輸出管理法でかかる開示が禁止あるいは規制されている場合、技術データをあらゆる国のあらゆる国民に対して開示すること (c) 技術データを、間接的か直接的かを問わず、輸出管理法で禁止されている、核兵器、ミサイル、化学兵器 / 生物兵器などの武器の最終用途のために輸出あるいは再輸出すること。

#### 10.9. 米国政府による規制

本ソフトウェアとユーザードキュメントは、該当するならば、アメリカ合衆国政府による FAR Section 12.212 および DFARS Section 227.7202 に定義されている「商用コンピュータソフトウェア」および「商用コンピュータソフトウェアドキュメント」と見な

されます。米国政府による本ソフトウェアとユーザードキュメントのいかなる使用、変更、再生、発表、実演、展示、公開は、本契約の諸条件に従ってのみ、行われるものとします。

#### 10.10. 準拠法；管轄権

本契約は、覚書を含む国際物品売買契約に係わる国際連合条約のあらゆる条項を除いて、米国カリフォルニア州法に準拠し、抵触法の原則は適用されません。本契約に係わる控訴は、カリフォルニア中央地区連邦裁判所あるいはカリフォルニアロサンゼルス地区州裁判所によって行われるものとします。両当事者は、統一コンピュータ情報取引法 (UCITA) が適宜、修正され、承認されるままに、ここに規定されている本ソフトウェアおよび保守サービスあるいは本契約に適用されることはないものとするに明白に同意します。

#### 10.11. 覚書

本契約は、書面に記載され、Parasoft およびお客様の役員によって署名されたもののみ、修正することができます。

#### 10.12. 存続

本契約の終了あるいは解除後も性質上存続すべき本契約の条項は有効に存続するものとします。かかる条項には、制限なく、責任制限、本ソフトウェアの権原および所有権に係わるすべての条項ならびにすべての一般条項が含まれます。

Copyright 1993-2017  
Parasoft Corporation  
101 East Huntington Drive, 2nd Floor  
Monrovia, CA 91016 USA

# 目次

---

## 概要

Static Analysis Engine (SAE) .....	1
Unit Test Connector (UTC) .....	1
Code Coverage Engine (CCE) .....	1

---

## はじめに

システム要件 .....	2
DTP Engine のインストール .....	3
ライセンスの設定 .....	4
DTP Server への接続 .....	4
ソース管理システム (SCM) への接続 .....	5

---

## Static Analysis Engine

基本的な解析 .....	8
テスト データの場所の指定 .....	8
詳細な進捗情報の表示 .....	9
エラー コードについて .....	9
テスト コンフィギュレーションの指定 .....	11
利用可能なテスト コンフィギュレーションの参照 .....	12
ビルトイン テスト コンフィギュレーション .....	12
カスタム ルールの作成 .....	14
テストスコープの定義 .....	15
リソース パターンの構文 .....	15
スコープの細かい調整 .....	17
プロジェクト ファイルの作成 .....	18
作成者の設定 .....	20
作成者設定の優先順位 .....	20
作成者の算定方法 .....	20
作成者 XML マップ ファイルの作成 .....	21

違反の抑制 .....	23
行単位の抑制 .....	23
重複コード解析 .....	25
フロー解析 .....	26
フロー解析の深さの設定 .....	26
タイムアウトの方針の設定 .....	27
メトリクス 解析 .....	29
メトリクスしきい値の設定 .....	29
IDE での DTP Engine の使用 .....	30

---

## レポート

レポートの出力場所の指定 .....	31
レポートのフォーマットの指定 .....	31
レポートの参照 .....	32
Development Testing Platform (DTP) Server への結果の送信 .....	38
DTP Server へのソース コードのパブリッシュ .....	38

---

## Unit Test Connector

JUnit テストの実行 .....	40
テストと開発成果物との関連付け .....	41
カバレッジの収集 .....	44

---

## Code Coverage Engine

単体テストのカバレッジ .....	46
カバレッジ データのマージ .....	46
アプリケーション カバレッジ .....	47
前提条件 .....	47
プロセスの概要 .....	47
カバレッジ収集のためのテスト対象アプリケーション (AUT) の構成 .....	47
テスト コンフィギュレーションと実行 .....	51
DTP への結果のアップロード .....	51

動的カバレッジ データ ファイルの生成と DTP へのアップロード .....	51
DTP でのカバレッジの参照 .....	52
Web アプリケーション カバレッジのチュートリアル .....	52

---

## DTP Engine for Java のカスタマイズ

現在の設定の参照 .....	61
変数の使用 .....	61
設定リファレンス .....	62

---

## 統合

ビルド システムとの統合 .....	89
Jtest 9.5 ユーザー向けの追加情報 .....	89
外部の解析ツールとの統合 .....	90
Checkstyle .....	90
FindBugs .....	91
ソース管理システムとの統合 .....	93
CI ツールとの統合 .....	94
Jenkins との統合 .....	94

---

## ヘルプの参照

テクニカル サポート .....	95
------------------	----

---

## サードパーティのコンテンツ

# 概要

Parasoft Development Testing Platform (DTP) Engine for Java は、生産性とソフトウェアの品質の向上に役立つさまざまなベスト プラクティスを自動化する統合ソリューションです。DTP Engine は、ソフトウェア品質ソリューション Parasoft Development Testing Platform ファミリーのコンポーネントです。DTP Engine がどのように Parasoft の Development Testing エコシステムに統合されるかについての詳しい情報は、次のマニュアルを参照してください。

『Parasoft Development Testing Solution』(PDF)

このドキュメントは、以下のエンジンの使用方法について説明します。

## Static Analysis Engine (SAE)

SAE は、静的解析やフロー解析など、実績のある品質プラクティスを使用して、コーディング ポリシーへの準拠を推進し、Java アプリケーションが期待どおりに動作することを保証します。詳細については 7 ページの「Static Analysis Engine」を参照してください。

## Unit Test Connector (UTC)

UTC を使用すると、オープン フォーマットのツールから単体テストを実行し、結果を Development Testing Platform (DTP) Server にレポートできます。詳細については 40 ページの「Unit Test Connector」を参照してください。

## Code Coverage Engine (CCE)

CCE は実行モジュールの実行中にカバレッジ情報を収集し、DTP Server に送信可能なレポートを生成します。詳細については 45 ページの「Code Coverage Engine」を参照してください。

# はじめに

この章は、DTP Engine の使用にあたって、お客様のシステムが要件を満たしているかを検証するのに役立ちます。また、すばやくコードの解析を開始できるよう、DTP Engine を構成するのに役立ちます。

## システム要件

### Windows 32-bit

- Windows 7、Windows 8
- 最小 4GB のメモリ \*
- 2GHz 以上のプロセッサ (x86 互換)、マルチ CPU 構成を推奨

### Windows 64-bit

- Windows 7 (x64)、Windows 8 (x64)、Windows 10、Windows Server 2008 (x64)、Windows Server 2012
- 最小 4GB のメモリ、8GB を推奨 \*
- 2GHz 以上のプロセッサ (x86\_64 互換)、マルチ CPU 構成を推奨

### Linux 32-bit

- Linux kernel 2.6 (またはそれ以降) および glibc 2.9 (またはそれ以降)
- 最小 4GB のメモリ \*
- 2GHz 以上のプロセッサ (x86 互換)、マルチ CPU 構成を推奨

### Linux 64-bit

- Linux kernel 2.6 (またはそれ以降) および glibc 2.12 (またはそれ以降)
- 最小 4GB のメモリ、8GB を推奨 \*
- 2GHz 以上のプロセッサ (x86\_64 互換)、マルチ CPU 構成を推奨

### Mac OS X 64-bit

- OS X 10.10 Yosemite、OS X 10.11 El Capitan
- 最小 4GB のメモリ、8GB を推奨 \*
- 2GHz 以上のプロセッサ (x86\_64 互換)、マルチ CPU 構成を推奨

Oracle 社の Java Runtime Environment を使用することを推奨します。他のフリーやオープンソースの実装を使用した場合、DTP Engines for Java が適切に機能しない可能性があります。

\*DTP Engines for Java は、32-bit マシンでは最大 1GB、64-bit マシンでは最大 2 GB までの RAM を Java Virtual Machine に割り当てることができます。[INSTALL\_DIR]/etc/jtestcli.jvm 構成ファイル (-Xmx オプション) を使用して、JVM プロセスに割り当てられるメモリの大きさを変更できます。

# DTP Engine のインストール

1. 任意のディレクトリにインストール パッケージを展開します。
2. 環境変数 `JAVA_HOME` が定義されていることを確認します。
3. [任意] Static Analysis Engine のディレクトリへのパスを `$PATH` に追加するか、`$PATH` に含まれるディレクトリに `jtestcli.exe` (Linux では `jtestcli`) をシンボリックリンクします。
4. [任意] Static Analysis Engine のディレクトリへのパスを指す `$JTEST_HOME` 環境変数を追加します。
5. [任意] Maven の `settings.xml` ファイルで次のプロパティを設定します。  
`pluginRepository, groupId, jtest.home`  
これらのプロパティの設定方法を参照するには、次の操作を行います。
  - a. `[INSTALL_DIR]manuals/plugins.html` を開きます。
  - b. `[JTEST MAVEN PLUGIN] > [使用方法] > [初期セットアップ]` を選択します。
6. [任意] `$ANT_HOME/lib` ディレクトリに `[INSTALL_DIR]/integration/ant/jtest-ant-plugin.jar` を格納します。Ant プラグインの jar の詳細なインストール方法を参照するには、次の操作を行います。
  - a. `[INSTALL_DIR]manuals/plugins-manual.html` を開きます。
  - b. `[JTEST ANT PLUGIN] > [使用方法] > [初期セットアップ]` を選択します。

## JVM、フレームワーク、アプリケーションのセットアップ

次の表は、DTP Engine のセットアップに使用する、その他の構成ファイルをまとめたものです。

ファイル	説明	ディレクトリ
<code>jtestcli.properties</code>	Static Analysis Engine プロパティのデフォルト設定が含まれます。	<code>[INSTALL_DIR]</code>
<code>jtestcli.jvm</code>	Java プロセスの起動時に <code>jtestcli.exe</code> ( <code>jtestcli</code> ) 実行モジュールが使用する JVM 引数が含まれます。	<code>[INSTALL_DIR]/etc</code>
<code>framework.properties</code>	起動された Felix OSGI フレームワークに渡されるプロパティが含まれます。通常は、このファイルを編集する必要はありません。	<code>[INSTALL_DIR]/etc</code>
<code>formatting.properties</code>	デフォルトの Static Analysis Engine プロパティのためのフォーマット ルールが含まれます。	<code>[INSTALL_DIR]/etc</code>
<code>logging.xml</code>	ログの構成ファイルです。デフォルトではサイレント コンソールを出力し、警告レベルの情報を <code>jtest.log</code> ファイルに出力します。	<code>[INSTALL_DIR]/etc</code>

ファイル	説明	ディレクトリ
logging.console.debug.xml	デバッグ レベルの情報をコンソールに出力するログの構成ファイルです。	[INSTALL_DIR]/etc

## ライセンスの設定

DTP Engine はローカル ライセンスまたはネットワーク ライセンスで動作します。ネットワーク ライセンスには以下の 2 種類があります。

- `ntp`: このライセンスは DTP に保存されます。解析できるファイル数の上限は、ライセンス契約で指定されたファイル数です。`license.use_network` に `true` が設定されている場合、このタイプがデフォルトです。
- `ls`: フローティングまたはマシン ロック ライセンスです。指定されたマシン数の上限まで利用できます。このライセンスは DTP の License Server に保存されます。

また、ネットワーク ライセンスには以下の 3 つのエディションがあり、利用できる機能が異なります。

- `desktop_edition`: デスクトップでの使用に最適な機能が利用できます。
- `server_edition`: サーバーのコマンド ライン モードでのパフォーマンスの高い使用に適した機能が利用できます。
- `custom_edition`: 利用可能な機能をカスタマイズできます。

## ローカル ライセンス

`.properties` 構成ファイルで以下を設定します。

1. `jtest.license.use_network` プロパティに `false` を設定します。
2. `jtest.license.local.password` プロパティにパスワードを設定します。

## マシン ID の取得

ローカル ライセンスを使用する場合、ライセンスを請求するためにマシン ID が必要です。マシン ID を取得するには、コマンド ライン ウィンドウで次のコマンドを実行します。

```
jtestcli -machineID
```

## ネットワーク ライセンス

`.properties` 構成ファイルで以下を設定します。

1. `jtest.license.use_network` プロパティに `true` を設定します。
2. `jtest.license.network.type` を設定します。
3. `jtest.license.network.edition` を設定します。

## DTP Server への接続

ライセンスの取得や、以下のようなチームワーク機能を拡張するには、DTP Server への接続が必要です。

- データベースへの解析結果のレポート (38 ページの「Development Testing Platform (DTP) Server への結果の送信」を参照)
- テスト コンフィギュレーションの共有
- 静的解析ルールの共有

[INSTALL\_DIR]\jtestcli.properties ファイルの以下の設定を変更し、DTP Server への接続を構成します。

```

ntp.server=[SERVER]
ntp.port=[PORT]
ntp.user=[USER]
ntp.password=[PASSWORD]

```

### 暗号化されたパスワードの作成

DTP Engine はパスワードを暗号化できます。パスワードを暗号化すると、DTP Server との通信のセキュリティが強化されます。暗号化されたパスワードを出力するには、次のコマンドを実行します。

```
-encodepass [MYPASSWORD]
```

コマンドから返された暗号化済みパスワードをコピーし、jtestcli.properties ファイルに貼り付けます。

```
ntp.password=[ENCODED PASSWORD]
```

## ソース管理システム (SCM) への接続

DTP Engine は任意のソース管理システムと統合できます。サポートされているのは以下の SCM です。

製品	テスト済みのバージョン
AccuRev	4.6、5.4、6.2
ClearCase	2003.06、7.0、8.0
CVS	1.1.2
Git	1.7
Mercurial	1.8.0 - 3.6.3
Perforce	2006、2012、2013、2014、2015
Serena Dimensions	9.1、10.1、10.3 (2009 R2)、12.2
Star Team	2005、2008、2009
Subversion (SVN)	1.2、1.3、1.4、1.5、1.6、1.7、1.8、1.9

製品	テスト済みのバージョン
Synergy/CM	6.4、7.0、7.1
Microsoft Team Foundation Server	2008、2010、2012、2013、2015
Visual SourceSafe	5.0、6.0、2005

SCM と接続するには、インストール ディレクトリにある `jtestcli.properties` ファイルを編集します。SCM 製品によってパラメーターが異なります。次のサンプルは、SVN への接続に必要なパラメーターを示しています。

```
scontrol.rep.type=svn
scontrol.rep.svn.url=https://svn_server/
scontrol.rep.svn.login=username
scontrol.rep.svn.password=password
scontrol.svn.exec=C:\\path\\to\\svn.exe
```

SCM との接続の構成については、60 ページの「DTP Engine for Java のカスタマイズ」を参照してください。

# Static Analysis Engine

Static Analysis Engine (SAE) は、静的解析やフロー解析など、実績のある品質プラクティスを使用して、コーディング ポリシーへの準拠を推進し、アプリケーションが期待どおりに動作することを保証します。以下のセクションでは、SAE を使用してコードを解析する方法について説明します。

- 基本的な解析
- テスト コンフィギュレーションの指定
- テストスコープの定義
- 重複コード解析
- IDE での DTP Engine の使用

# 基本的な解析

DTP Engine for Java を起動するには、コマンドラインから `jtestcli.exe` を実行します。少なくともテスト コンフィギュレーションおよび解析対象コードのデータ ファイルの場所を指定する必要があります。詳細については 11 ページの「テスト コンフィギュレーションの指定」および 8 ページの「テスト データの場所の指定」を参照してください。

```
jtestcli.exe -settings settings.properties -config "builtin://Recommended Rules"
-data demo.data.json -report report
```

サポートされているビルド システム (Maven、Ant、および Gradle) の専用プラグインから `jtestcli.exe` を起動することもできます。ビルド システム用プラグインの詳細については 89 ページの「ビルド システムとの統合」を参照してください。

プラグインからの実行時に `jtestcli.exe` に渡される引数を表示するには、環境変数 `$XTEST_ITX=714` を追加します。

## テスト データの場所の指定

`-data` スイッチを使用して、ビルドシステム プラグインによって生成される `*.json` データファイルのディレクトリを指定します。自動生成された `*.json` データ ファイルは、ワークスペースと同じ情報を Static Analysis Engine 固有のフォーマットで表します。

```
-data file
```

ビルド システム プラグインは、`data` 引数を自動的に追加し、生成された値をインクルードします。そのため、これをプラグインの構成に追加する必要はありません。

## プロパティ設定のパターン

```
jtest.data=[path to file]
```

## データ ファイルのフォーマット

データ ファイル (`jtest.data.json`) は、Static Analysis Engine のビルド システム プラグインによって自動生成され、フォーマットされた json オブジェクトのストリームです。データ ファイルは、テスト対象プロジェクトに関してワークスペースと同じ情報を提供します。現行の Static Analysis Engine は、`classpath_project` 共通オブジェクト型をサポートします。サンプルについては 18 ページの「プロジェクト ファイルの作成」を参照してください。

## データ ファイル オブジェクトのサンプル

```
{
  "type": "classpath_project",
  "name": "project_name",
  "location" "/absolute/path/to/project",
  "compilations": [
    {
      "sourcepath": [
        "/absolute/path/to/srcdir1"
      ],
      "classpath": [
        "/absolute/path/to/classdir1",
        "/absolute/path/to/buildfile.jar"
      ],
      "bootpath": [
        "absolute/path/to/java/ajavalib.jar"
      ],
      "encoding": "project_encoding_name",
      "sourcelevel" "project_source_level"
    }, {
      ...
    }
  ]
}
{
  ...
}
```

## 詳細な進捗情報の表示

詳細な進捗情報を表示するには、`-showdetails` スイッチを使用します。

```
-showdetails
```

### Ant および Maven のパターン

```
<showdetails>true</showdetails>
```

### プロパティ設定のパターン

```
console.verbosity.level=high
```

## エラーコードについて

Static Analysis Engine は、以下の場合にエラー終了コード (0 以外の値) を返します。

- ライセンスがない状態で Static Analysis Engine が実行された。
- 内部例外が発生し、テスト プロセスが終了した。
- コマンドラインの書式が誤っているか、存在しないリソースを参照している。15 ページの「テストスコープの定義」を参照。

`-fail` オプションを使用すると、静的解析が違反をレポートした場合もエラーコードを確認することができます。

# テスト コンフィギュレーションの指定

テスト コンフィギュレーションは、DTP Engine がどのようにコードを解析するかを定義したものです。たとえば、有効にする静的解析ルール、実行するテストなどの解析パラメーターを定義します。DTP Engine にはビルトイン テスト コンフィギュレーションがあらかじめ用意されていますが、ユーザーは、DTP Server でユーザー独自のテスト コンフィギュレーションを作成して保存することもできます。DTP プラグインから DTP Server にアクセスできます。DTP Report Center の管理者レベルのアクセス権を持っている場合、DTP で直接テスト コンフィギュレーションを作成できます ([管理] > [エンジン] > [テスト コンフィギュレーション])、

DTP Server からユーザー定義のテスト コンフィギュレーションをダウンロードし、\*.properties ファイルとして [INSTALL\_DIR]/configs/user ディレクトリに保存できます。

実行するテスト コンフィギュレーションを指定するには、-config スイッチを使用します。

```
-config "user://Configuration Name"
```

## Ant および Maven のパターン

```
<config>user://Configuration Name</config>
```

## プロパティ設定のパターン

```
jtest.config=user://Configuration Name
```

実行するテスト コンフィギュレーションを指定する方法はいくつかあります。

(デフォルトでは、builtin://Recommended Rules テスト コンフィギュレーションが使用されます)

## ビルトイン コンフィギュレーション

```
-config "builtin://Recommended Rules"
```

## ユーザー定義コンフィギュレーション

```
-config "user://Foo Configuration"
```

## DTP Server にホストされたコンフィギュレーション

```
-config "dtp://Foo Team Configuration"
-config "dtp://FooTeamConfig.properties"
```

ファイル名または URL を使用してテスト コンフィギュレーションを参照することもできます。

## ファイル名による指定

```
-config "C:\Devel\Configs\FooConfig.properties"
```

## URL による指定

```
-config "http://foo.bar.com/configs/FoodConfig.properties"
```

## 利用可能なテスト コンフィギュレーションの参照

利用可能なテスト コンフィギュレーションを出力するには、`-listconfigs` スイッチを使用します。引数を指定してコンフィギュレーションをフィルターできます。ワイルドカード "\*" を使用できます。

<code>-listconfigs</code>	利用可能なすべてのコンフィギュレーションを表示
<code>-listconfigs builtin</code>	すべてのビルトイン コンフィギュレーションを表示
<code>-listconfigs builtin*Secure*</code>	名前に "secure" が含まれるすべてのビルトイン コンフィギュレーションを表示

```
-listconfigs
```

## ビルトイン テスト コンフィギュレーション

次の表は、DTP Engine の `[INSTALL]/configs/builtin` ディレクトリにインストールされるビルトイン テスト コンフィギュレーションです。

ディレクトリ://コンフィギュレーション名	説明
Recommended Rules	推奨ルールのデフォルト コンフィギュレーションです。重要度 1 および 2 の大部分のルールをカバーします。Flow Analysis Fast コンフィギュレーションのルールを含みます。
Find Duplicated Code	重複コードをレポートする静的コード解析ルールを適用します。重複コードは、アプリケーションの設計が適切でないことを表す可能性があり、保守性の問題につながる場合があります。
Internationalize Code	国際化を妨げる可能性が高いコードを検出する静的コード解析を適用します。
Juliet 1.1 2011	
Metrics	コード メトリクスを計測します。
New Features in JDK 1.5	

ディレクトリ :// コンフィギュレーション名	説明
New Features in JDK 7	
DISA-STIG for Java	DISA-STIG 標準で識別された問題を検出するルールを含みます。
Critical Rules	重要度 1 のルールの大部分および Flow Analysis Fast コンフィギュレーションのルールを含みます。
Flow Analysis	デフォルトの設定を使用したフロー解析ルールです。
Flow Analysis Agressive	深いフロー解析を行うルールを含みます。このコンフィギュレーションの実行には、非常に長い時間がかかる可能性があります。
Flow Analysis Fast	浅いフロー解析を行うルールを含みます。
Demo Configuration	さまざまなコード解析技術をデモンストレーションするルールを含みます。大規模なコードの解析には適していない可能性があります。
Find Memory Problems	メモリ管理の問題を検出するルールを含みます。
Find Unused Code	未使用のコード / デッド コードを識別するルールを含みます。
CWE-SANS Top 25 [2009/2011]	CWE-SANS の Top 25 Most Dangerous Programming Errors として分類された問題を検出するルールを含みます。
SAMATE NIST 2010	NIST SAMATE 標準で識別された問題を検出するルールを含みます。
OWASP Top 10 [2007, 2010, 2013]	OWASP のセキュリティ脆弱性トップ 10 で識別された問題を検出するルールを含みます。
PCI Data Security Standard	PCI Data Security Standard で識別された問題を検出するルールを含みます。
Thread Safe Programming	マルチスレッド環境で実行するのが危険なコードを検出し、デッドロック、競合状態、通知の失敗、無限ループ、データ破壊などのスレッド処理でよくある問題を防ぐのに役立つルールです。
Unit Tests	生成されるレポート ファイルに、単体テストの実行データを含めます。

ディレクトリ :// コンフィギュレーション名	説明
Calculate Application Coverage	アプリケーション カバレッジ データを処理して coverage.xml file を生成します。詳細については 47 ページの「アプリケーション カバレッジ」を参照してください。
Unit Testing Best Practices	単体テストのベスト プラクティスを徹底するのに役立つほか、単体テストでアサーションが確実に作成されるようにします。
Code Smells	Kent Beck と Martin Fowler による Code Smells ドキュメント ( <a href="http://xp.c2.com/CodeSmell.html">http://xp.c2.com/CodeSmell.html</a> ) に基づいたルール。
TDD	TDD (Test Driven Development) コンフィギュレーションは、Code Smells ドキュメント ( <a href="http://xp.c2.com/CodeSmell.html">http://xp.c2.com/CodeSmell.html</a> ) に基づいたルール、テスト対象クラスに対して JUnit テスト クラスが包括的かどうかをチェックするルール、および Critical Rules (Must Have) テスト コンフィギュレーションのルールを含みます。

## カスタム ルールの作成

カスタム ルールを作成するには、RuleWizard を使用します。Static Analysis Engine でルールを使用するには、テスト コンフィギュレーションでルールを有効化し、カスタム ルール ファイルを以下のどちらかのディレクトリに配置する必要があります。

- [INSTALL\_DIR]\rules\user\
- [DOCUMENTS DIR]\Parasoft\[engine]\rules  
[DOCUMENTS DIR] は Windows の "My Documents" ディレクトリを指します。

# テストスコープの定義

テスト データのディレクトリやテスト対象のファイルまたはファイルのセットを入力スコープとして指定して解析することもできます。テスト対象のファイルまたはファイルのセットを指定するには、`-resource` スイッチを使用します（データ ファイルの場所の指定については 8 ページの「テスト データの場所の指定」を参照）。複数の `-resource` スイッチを使用して、複数のリソースを指定することもできます。

```
-resource [pattern]
```

## Ant および Maven のパターン

```
<resource>pattern</resource>
```

`<resources>` 要素を使用した複数のリソースの定義

```
<resources>
  <resource>pattern1</resource>
  <resource>pattern2</resource>
</resources>
```

## プロパティ設定のパターン

```
jtest.resource=pattern
```

複数のリソースをカンマで区切ります。

```
jtest.resource=pattern1,pattern2
```

## リソース パターンの構文

リソース パターンは、次の規則に従って生成されたパスと比較されます。:

```
ProjectName/resource/inside/of/project/directory.extension
ProjectName/EXT/resource/inside/of/external/source/directories.extension
```

`ProjectName` の値は、次のルールに従って生成されます。

- Ant のプロジェクト名は `project` タグの `name` プロパティから 取得されます。
- Maven のプロジェクト名は `groupId:artifactId` から 取得されます。
- Gradle のプロジェクト名は `gradle.settings` ファイルから 取得されます。
- IDE で設定されたプロジェクト名はそのまま使用されます。

`projectNameTemplate` パラメーターを使って、Maven と Gradle のプロジェクト名をカスタマイズできます。詳細については以下の場所にあるプラグイン マニュアルを参照してください。

[INSTALL]/manuals/plugins-manual.html

指定したスコープが外部リソースを含む場合、生成されるパスに文字列 `EXT` が含まれます。たとえば、解析対象のプロジェクトが、別の場所にあるソース ファイルをリンクで含んでいるかもしれません。以下の例では、`Simple.java` ファイルは `Demo` プロジェクトにあります、`Money.java` ファイルはリンクで追加されていてプロジェクトの外部にあります。

```
Demo/src/examples/eval/Simple.java
Demo/EXT/examples/bank/Money.java
```

Ant 形式のワイルドカードやその他のパラメーターを使用して柔軟にパターンを構成できます。次の表は、サポートされているパラメーターの説明です。

パラメーター	説明
?	パス区切り文字以外の任意の 1 文字に一致するワイルドカード
*	パス区切り文字以外の任意の 0 文字以上に一致するワイルドカード
** (	0 個以上のパス セグメントに一致するワイルドカード
/	すべてのオペレーティング システムの区切り文字
"[non-alphanumeric characters]"	リソース パスに空白またはその他の非英数文字が含まれる場合、引用符を使用します。

`-resource` 引数に 1 つしか値が含まれていない場合、値はプロジェクト名と比較され、ワイルドカード (\*) が使用されます。次の表は、`-resource` スイッチの使用法のサンプルです。

式	結果
<code>-resource ProjectName/src</code>	指定されたディレクトリおよびサブディレクトリのすべてのファイルが解析されます (ワイルドカードを使用しない場合、ディレクトリの完全な名前を指定する必要があります)。
<code>-resource ProjectName/**/*.java</code>	指定されたプロジェクトのすべての Java ファイルが解析されます。
<code>-resource **/src/main/java/my/company/*.java</code>	現在のビルドに含まれるすべてのプロジェクトの指定したサブディレクトリにあるすべての Java ファイルが解析されます。
<code>-resource ProjectName/src/main/java/my/company/File.java</code>	指定された単一のファイルが解析されます。

式	結果
<code>-resource c:/resource.lst</code>	resource.lst ファイルに記載されたプロジェクトを解析します。以下のように 1 行に 1 個のプロジェクト名を指定してください。 ProjectName1 ProjectName2  -resource 引数の値として、resource.lst ファイルのパスを指定します。

## スコープの細かい調整

スコープにその他のフィルターを適用するには、`-include` および `-exclude` スイッチを使用します。

- `-include` スイッチは、ファイル システムのパスに一致するファイルだけをテストするよう指定します。その他のファイルはスキップされます。
- `-exclude` スイッチは、ファイル システムのパスに一致するファイル以外をテストするよう指定します。

`-include` と `-exclude` の両方のスイッチが指定されている場合、`-include` に一致するが、`-exclude` に一致しないパターンだけがテストされます。

```
-include pattern
-exclude pattern
```

## Ant および Maven のパターン

```
<include>pattern</include>
<exclude>pattern</exclude>
```

## プロパティ設定のパターン

```
jtest.include=pattern
jtest.exclude=pattern
```

`-include` および `-exclude` フィルターには、Ant 形式のワイルドカードやその他のパラメーターを使用できます。次の表は、使用方法の説明です。

パラメーター	設定
?	パス区切り文字以外の任意の 1 文字に一致するワイルドカード
*	パス区切り文字以外の任意の 0 文字以上に一致するワイルドカード
**	0 個以上のパス セグメントに一致するワイルドカード

パラメーター	設定
/	セパレーター
path:	ハード ディスクの絶対パスに一致する接頭辞

多数のファイルを `-include/-exclude` で指定する場合、リスト ファイル (\*.lst) を作成し、そのリスト ファイルのパスを指定することもできます。\*.lst ファイルの各アイテムが個別のエントリとして扱われます。

### サンプル

構文	説明
<code>-include com/**</code>	"com" で始まるすべてのパッケージをテストします。
<code>-include **/Bank.java</code>	BankAccount.java ファイルだけをテストします。
<code>-include path:C:/Project/src/**</code>	C:/Project/src のすべてのサブファイルおよびサブディレクトリをテストします。
<code>-include path:C:/Project/src/*</code>	C:/Project/src のすべてのファイルをテストしますが、サブディレクトリはテストしません。
<code>-include c:/include.lst</code>	include.lst ファイルにリストされたすべてのファイルをテストします。各行が 1 つのパターンとして扱われます。例：  include.lst ファイルに次の行がある場合、 <code>**/*Account</code> <code>path:**/Bank.java</code> コマンドで次のように指定したのと同じです。 <code>-include **/*Account</code> <code>-include path: **/Bank.java</code>
<code>-exclude **/internal/**</code>	パッケージ名に "internal" を含むクラス以外のすべてのものをテストします。

## プロジェクト ファイルの作成

`-project.*` スイッチを使用すると、プロジェクト ファイルを作成できます。プロジェクト ファイルには、DTP Engine の実行に必要な情報が含まれています。ソリューションがない場合に `-project` スイッチを使用します。このスイッチを複数回使用して、多数のプロジェクトを解析できます。Ant 形式のワイルドカードもサポートされているほか、\*.lst ファイルへのパスも指定できます。次の表は、`-project` スイッチに使用できるパラメーターの説明です。

パラメーター	説明
<code>-project.location</code>	プロジェクトファイルの生成先を指定します。 <code>-project.location</code> オプションが指定されていない場合、他のすべての <code>-project.*</code> オプションが無視されます。

パラメーター	説明
-project.name	プロジェクト名を設定します。
-project.encoding	プロジェクトのエンコーディングを設定します。
-project.sourcepath	プロジェクトのソースを設定します。複数のソースフォルダを指定する場合、このオプションを複数回使用します。
-project.sourcelevel	Java コンパイラの準拠レベルを指定します。
-project.classpath	classpath を指定します。
-project.javahome	ライブラリを検索し bootclasspath に追加するために使用します。
-project.classpath.jars.dir	プロジェクトのクラスパスに追加しなければならない jar ファイルがあるディレクトリへのパスを指定します。json ファイルで jar ファイルの順番をチェックし、プロジェクトの要件に順番が合っていることを確認してください。
-project.junit.outcomes	レポートを生成するために使用する、JUnit テストの結果がある XML ファイルへのパスを指定します。複数のレポート ファイルを指定するには、このオプションを複数回指定してください。
-project.compilation.classes	Calculate Application Coverage コンフィギュレーションを実行するために必要なメタデータ情報を生成するために使用する、コンパイル済みプロジェクト クラスへのパスを指定します。

## サンプル

```
-project.location C:/ExampleProject
-project.name ExampleProject
-project.encoding UTF-8
-project.sourcepath C:/ExampleProject/src
-project.sourcelevel C:/ExampleProject/src-test
-project.classpath C:/ExampleProject/lib/test.jar;C:/ExampleProject/lib/test2.jar
-project.javahome C:/Program Files/Java/jdk1.7
```

# 作成者の設定

解析中に作成者に関する情報を収集するよう DTP Engine を設定すると、タスクの割り当てが容易になります。作成者情報を DTP サーバーに送信すると、Process Intelligence Engine (PIE) などの他の解析コンポーネントがその情報を利用できるため、欠陥の修正や開発プロセスの最適化に役立ちます。

ソース管理システムの情報、ソースを作成者に直接マッピングする XML ファイル、現在のローカルユーザーに基づいて作成者を割り当てるよう DTP Engine を設定できます。また、@author Javadoc タグを使用することもできます。

## 作成者設定の優先順位

作成者の優先順位は、.properties 設定ファイルの設定を上から下へ読み取ることによって決定されます。複数の作成者のソースが使用されている場合、次の優先順位が適用されます。

1. ソース管理システムの情報
2. @author タグ
3. XML マップ ファイル
4. 現在のユーザー

選択されたオプションのいずれかで作成者を決定できない場合（たとえば @author タグが選択されているが、ファイルに @author タグがない場合）、選択された次のオプションに基づいて作成者が決定されます。最終的に作成者を決定できない場合、ユーザーは「不明」に設定されます。同様に、どのオプションも選択されていない場合もユーザーは「不明」に設定されます。

## 作成者の算定方法

作成者の決定方法を指定するには、jtestcli.properties 設定ファイルを編集します。

```
scope.local=[true or false]
scope.scontrol=[true or false]
scope.xmlmap=[true or false]
```

- scope.scontrol を true に設定すると、ソース管理システムのチェックイン データに基づいて作成者が算定されます。このオプションは、デフォルトでは false です。
- scope.local を true に設定すると、ローカル ユーザーおよびシステム ファイル更新日時に基づいて作成者が算定されます。このオプションは、デフォルトでは true です。
- scope.xmlmap を true に設定すると、XML マップ ファイル（書式については 21 ページの「作成者 XML マップ ファイルの作成」を参照）に保存された情報に基づいて作成者が算定されます。このオプションは、デフォルトでは false です。
- すべてのプロパティに true が設定されている場合、scope.scontrol が優先されます。

## その他の作成者設定

デフォルトでは大文字と小文字が区別されますが、大文字と小文字の区別を無効にすることができます。

```
authors.ignore.case=true
```

`authors.user[identiflyer]` オプションを使用してユーザー名、E-mail アドレス、およびユーザーのフルネームを設定できます。

例 :

```
authors.user1=john, john.doe@company.com, John Doe
```

ユーザーがチームからいなくなった場合や、作成者を他のユーザーに変更する必要がある場合、`authors.mapping[x,y]` オプションを使用します。

```
authors.mapping1=old_user,new_user
```

作成者を他のユーザーに変更する場合、作成者対作成者のマッピング情報をローカルまたは共有 XML マップ ファイルに保存できます。

```
authors.mappings.location=[local or shared]
```

マッピング ファイルを共有する場合、共有 XML ファイルの場所を指定します。

```
authors.shared.path=[path to file]
```

## 作成者 XML マップ ファイルの作成

`<authorship>` 要素はマッピング 情報の開始を表します。

`<file />` 要素は `<authorship>` 要素内に置かれ、`author` および `path` の 2 つのプロパティでユーザーをファイルまたはファイルのセットにマッピングします。

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE authorship (View Source for full doctype...)>
  <authorship>
    <!-- assigns all files named: "foo/src/SomeClass.java" to "author1" -->
    <file author="author1" path="foo/src/SomeClass.java
  " />
```

ワイルドカードを使用して作成者をファイルのセットにマッピングできます。次の表はワイルドカードの使用例です。

ワイルドカード表現	説明
?oo/src/ SomeClass.java	名前が / 以外の文字で始まり "oo/src/" で終わるすべてのファイルを割り当てます。
** .cs	任意のディレクトリにあるすべての *.cs ファイルを割り当てます。
**/src/**	パスに "src" という名前のフォルダーを含むすべてのファイルを割り当てます。
src/**	ディレクトリ "src" にあるすべてのファイルを割り当てます。
src/**/Test*	"src" ディレクトリ内にあり、名前が "Test" で始まるすべてのファイルを割り当てます ("src/some/other/dir/TestFile.c" など)。

**マッピングの順序の問題** マッピング ファイルは上から下に読み取られるため、より詳細なマッピングを上配置すると、正しいファイルに作成者がマッピングされるようになります。

# 違反の抑制

違反を抑制すると、DTP Engine は特定の静的解析タスク（1 つのルールに対して複数のタスクがレポートされる場合があります）をそれ以降レポートしなくなります。違反の抑制は、特定のルール違反に対して繰り返しメッセージを受け取りたくない場合に便利です。特定のルールのすべての違反メッセージを受け取りたくない場合は、テスト コンフィギュレーションでルールを無効にします。

IDE で DTP Engine を使用している場合、GUI を使用して抑制を定義できます（詳細についてはお使いの IDE 用 DTP プラグインのマニュアルを参照してください）。IDE で DTP Engine を使用していない場合は、次の構文を使用してソース コード内に抑制を定義します。

## 行単位の抑制

```
<suppression keyword> [<rule category> | <rule category> . <rule id> | <rule category > - <rule severity> | ALL ] <suppression comment>
```

### 行単位の抑制の例

```
// parasoft-suppress CODSTA "suppress all rules in category CODSTA"

// parasoft-suppress CODSTA.NEA "suppress rule CODSTA.NEA"

// parasoft-suppress CODSTA-1 "suppress all rules in category CODSTA with severity level 1"

// parasoft-suppress ALL "suppress all rules"

// parasoft-suppress CODSTA FORMAT.MCH JAVADOC-3 "suppress all rules in category CODSTA and rule FORMAT.MCH and all rules in category JAVADOC with severity level 3"
```

## ブロック単位の抑制

```
<begin suppression keyword> [<rule category> | <rule category> . <rule id> | <rule category > - <rule severity> | ALL ] <suppression comment>
```

```
..... source code block .....
```

```
<end suppression keyword> [<rule category> | <rule category> . <rule id> | <rule category > - <rule severity> | ALL ] <suppression comment>
```

### ブロック単位の抑制の例

```
// parasoft-begin-suppress CODSTA "begin suppress all rules in category CODSTA"
.....
// parasoft-end-suppress CODSTA "end suppress all rules in category CODSTA"

// parasoft-begin-suppress CODSTA.NEA "begin suppress rule CODSTA.NEA"
```

```
.....
// parasoft-end-suppress CODSTA.NEA "end suppress rule CODSTA.NEA"

// parasoft-begin-suppress CODSTA-1 "begin suppress all rules in category CODSTA
with severity level 1"
.....
// parasoft-end-suppress CODSTA-1 "end suppress all rules in category CODSTA with
severity level 1"

//parasoft-begin-suppress ALL "begin suppress all rules"
.....
// parasoft-end-suppress ALL "end suppress all rules"

// parasoft-begin-suppress CODSTA FORMAT.MCH "begin suppress all rules in category
CODSTA and rule FORMAT.MCH"
.....
// parasoft-end-suppress CODSTA FORMAT.MCH "end suppress all rules in category
CODSTA and rule FORMAT.MCH"

// parasoft-begin-suppress CODSTA "begin suppress all rules in category CODSTA"
.....
// parasoft-end-suppress CODSTA-1 "end suppress all rules in category CODSTA with
severity level 1; however rules with severity level 2-5 in category CODSTA are still
suppressed."
.....
// parasoft-end-suppress CODSTA "end suppress all rules in category CODSTA"

// parasoft-begin-suppress ALL "begin suppress all rules"
.....
// parasoft-end-suppress CODSTA FORMAT-1 "end suppress all rules in category CODSTA
and all rules in category FORMAT with severity level 1; however, others rules in
CODSTA and FORMAT-1 are still suppressed."
.....
// parasoft-end-suppress ALL "end suppress all rules"

//parasoft-begin-suppress ALL "begin suppress all rules, since no end suppression
comment, all rules will be suppressed starting from this line"
```

# 重複コード解析

DTP Engine は重複コードをチェックできます。重複コードはアプリケーションの設計が不適切であることを表す場合があります、保守のコストが増大する可能性があります。重複コードの解析では、コードが小さな言語要素（トークン）に分解されます。重複と見なすコードを定義したルールのセットに従ってトークンが分析されます。トークンを解析するルールには次の 2 つのタイプがあります。

- 文字列リテラルなど、単一のトークンの重複を発見するための単純なルール
- 重複メソッドやステートメントなど、複数のトークンの重複を発見するための複合ルール

重複コード検出ルールを実行するには、Find Duplicated Code テスト コンフィギュレーションを実行します。

```
builtin://Find Duplicated Code
```

# フロー解析

フロー解析は、静的解析テクノロジーの一種です。アプリケーションの実行パスのシミュレーションをはじめ、複数の解析手法を利用して実行時のバグを引き起こす可能性があるパスを特定します。フロー解析が検出するバグには、未初期化メモリの使用、NULL ポインターの間接参照、ゼロによる除算、メモリ リーク、リソース リークなどが含まれます。

フロー解析は複雑なパスを特定してトレースするため、人間によるテストや検査では発見が難しく、静的解析や単体テストでも発見されないことが多いバグを検出します。

コードを実行せずにバグを検出するフロー解析の機能は、特にレガシー コード ベースや組込みコード（つまりそのようなエラーの実行時検出が効果的ではなかったり不可能だったりするコード）に有効です。

フロー解析ルールを実行するには、解析にフロー解析用テスト コンフィギュレーションのいずれかを実行します。

```
builtin://Flow Analysis Fast
builtin://Flow Analysis Standard
builtin://Flow Analysis Aggressive
```

## フロー解析の深さの設定

フロー解析エンジンは、解析されたコードからパスを分析し、さまざまな問題を検出します。アプリケーション全体にわたって「実行される可能性があるパス」をすべて解析するのは現実的ではない場合もあるため、解析レベルの深さを設定できます。解析レベルが深いほど、発見される違反の数が増えます。ただし、パフォーマンスが低下し、若干メモリの消費量が増加します。

解析の深さは、以下の方法で指定できます。

- DTP でテスト コンフィギュレーション インターフェイスを使用します。Report Center > [ テスト コンフィギュレーション ] > [ 静的解析 ] > [ フロー解析詳細設定 ] > [ パフォーマンス ] > [ 解析の深さ ] にアクセスし、以下のいずれかのオプションを選択します。
  - **最も浅く (最も高速)**: ソース コード中の最も明らかな問題だけを発見します。発見する対象は、問題が発生しているコード箇所に「問題の起点」が近い場合だけに制限されます。発見される違反の実行パスは通常、1 つの関数のいくつかの行にわたります。4 以上の関数呼び出しにわたることはまずありません。
  - **浅く (高速)**: [ 最も浅く ] オプションと同様、ソース コード中の最も明らかな問題だけを発見します。ただし、[ 浅く ] オプションは [ 最も浅く ] オプションよりも多くの問題を検出し、より長いパスを検証できます。
  - **標準**: 何十もの要素を含む実行パスにおける、多くの複雑な問題を発見します。[ 標準 ] オプションは、[ 浅く ] オプションよりも深い解析を実行するのに加えて、より複雑な問題を探します。たとえば、「不正なフローのために 1 つの関数で発生する問題」や、「解析対象プロジェクト中の異なる箇所にある異なる関数間で不適切なやり取りがあるために発生する問題」などです。多くの場合、[ 標準 ] オプションが発見する違反は、解析対象ソース コード中の重要なバグを明らかにし、そのコード行は何十行にも及びます。

- **深く (低速):** [標準] オプションで定義されているのと同等の複雑さと性質の問題を、より多く検出できます。ただし、解析速度は [標準] オプションよりも遅くなります。
- **徹底的 (より低速):** より複雑な問題を発見します。コードベースを徹底的に解析するため、時間を必要とします。しかし、アプリケーション中の異なる場所にある何百行というコードに違反パスがわたるような、非常に複雑な問題を数多く発見します。このオプションは夜間のテスト実行で使用することを推奨します。
- テスト コンフィギュレーション ファイルを手動で編集します。エディターでテスト コンフィギュレーション ファイルを開き、  
`com.parasoft.xtest.checker.flowanalysis.depthOfAnalysis` プロパティの値を設定して解析レベルの深さを指定します。値は以下のとおりです。  
0 - 最も浅く、1 - 浅く、2 - 標準、3 - 深く、4 - 徹底的

デフォルトの解析レベルの深さは [標準] です。

## タイムアウトの方針の設定

解析レベルの深さの他に、フロー解析エンジンはタイムアウト設定を利用して、妥当な時間内で解析が終了するようにします。以下の方法でタイムアウトの設定を行うことができます。

- DTP でテスト コンフィギュレーション インターフェイスを使用します。Report Center > [テスト コンフィギュレーション] > [静的解析] > [フロー解析詳細設定] > [パフォーマンス] > [タイムアウトの方針] にアクセスし、以下のいずれかのオプションを選択します。
  - **時間:** 指定時間が過ぎると、指定したホットスポットの解析が停止します。注意：このオプションを選択すると、レポートされる違反件数がやや不安定になる場合があります。
  - **インストラクション:** フロー解析エンジンのインストラクションの実行回数が指定の回数を超えると、指定したホットスポットの解析が停止します。注意：環境に合った適切なインストラクション数を決定するには、生成されたレポートの [セットアップの問題] セクションでタイムアウト情報を確認してください。
  - **オフ:** タイムアウトしません。注意：このオプションを選択すると、解析が終了するまで時間がかかる場合があります。
- テスト コンフィギュレーション ファイルを手動で編集します。エディターでテスト コンフィギュレーション ファイルを開き、  
`com.parasoft.xtest.checker.flowanalysis.hotSpotTimeoutStrategy` プロパティの値を設定します。
  - **time:** 指定時間が過ぎると、指定したホットスポットの解析が停止します。  
`com.parasoft.xtest.checker.flowanalysis.hotSpotTimeoutSeconds` プロパティでタイムアウトの秒数を指定します。
  - **ticks:** フロー解析エンジンのインストラクションの実行回数が指定の回数を超えると、指定したホットスポットの解析が停止します。  
`com.parasoft.xtest.checker.flowanalysis.hotSpotTimeoutTicks` プロパティで、インストラクションの数を指定します。注意：環境に合った適切なインストラクション数を決定するには、生成されたレポートの [セットアップの問題] セクションでタイムアウト情報を確認してください。
  - **off:** タイムアウトなし。

デフォルトのタイムアウト オプションは [時間] であり、タイムアウト秒数は 60 秒です。解析中に発生したフロー解析タイムアウトについての情報を得るには、解析後に生成されるレポートで [セットアップの問題] セクションを確認してください。

# メトリクス 解析

DTP Engine はコードの複雑性、オブジェクト間の結合、凝集性の欠如などのコードの潜在的な弱点を理解するのに役立つコード メトリクスを計測できます。メトリクス解析ルールを実行するには、Metrics テスト コンフィギュレーションを実行します。

```
builtin://Metrics
```

メトリクス解析の結果は、DTP Engine が生成する HTML および XML レポートファイルに追加されます。レポートの詳細については 35 ページの「メトリクス サマリー」を参照してください。

## メトリクスしきい値の設定

上限または下限を指定して、メトリクスが指定された値範囲を超えた場合に静的解析違反をレポートすることができます。たとえば、論理行数を制限する場合、論理行数メトリクスが制限を超えた場合に違反をレポートするよう、Metrics テスト コンフィギュレーションを設定することができます。

DTP Engine に付属している Metrics テスト コンフィギュレーションには、デフォルトのしきい値が設定されています。ファイル数 (METRIC.NOF) など、しきい値を設定できないルールもあります。

メトリクスしきい値を設定するには、次の方法があります。

- DTP のテスト コンフィギュレーション インターフェイスを使用する（詳細については『Development Testing Platform ユーザーズ ガイド』の「Report Center」>「テスト コンフィギュレーション」>「テスト コンフィギュレーションの編集」>「[メトリクス] タブ」を参照）
- IDE のインターフェイスを使用してテスト コンフィギュレーションを編集する（使用している IDE に応じた『Parasoft DTP プラグイン ユーザーズ ガイド』の「テスト コンフィギュレーション」>「カスタムテスト コンフィギュレーションの作成」を参照）
- テスト コンフィギュレーション ファイルを手動で編集する
  1. Metrics ビルトイン コンフィギュレーション ([INSTALL]/configs/builtin) をユーザー コンフィギュレーション ディレクトリ (([INSTALL]/configs/user) に複製します。
  2. 複製されたコンフィギュレーションをエディターで開き、[METRIC.ID].ThresholdEnabled プロパティに true を設定します。
  3. 次の書式に従って [METRIC.ID].Threshold プロパティに上限および下限を指定します。  
[METRIC.ID].Threshold=1 [lower boundary value] g [upper boundary value]
  4. テスト コンフィギュレーションを保存し、カスタム メトリクス テスト コンフィギュレーションを使用して解析を実行します。

# IDE での DTP Engine の使用

IntelliJ IDEA、NetBeans または Eclipse 内で DTP Engine を使用できます。IDE との統合機能は、ローカルでコードの静的解析を実行し、結果を参照したり、DTP サーバーに格納されたデータやテスト コンフィギュレーションを利用するためのデスクトップ インターフェイスを提供します。また、DTP Server から開発環境に指摘事項をインポートすることもできます。

このような統合機能は、DTP Plugin for IntelliJ IDEA、NetBeans、Eclipse および DTP Engine Plugin によって実現されます。インストールや使用方法などの詳細については、IDE のメニュー バーから [Parasoft] > [オンライン ヘルプ] を選択するか、製品に付属の『Jtest デスクトップ ユーザーズ ガイド』を参照してください（オンライン ヘルプの参照を推奨します）。

# レポート

デフォルトでは、DTP Engine は結果をコンソール出力に表示するほか、`[WORKING_DIR]/reports` ディレクトリに HTML レポートを保存します。HTML レポートの元になるデータは XML ファイルとしてディレクトリに保存されます。この XML ファイルを使用すると、サポート対象の IDE 用 Parasoft DTP Plugin および Parasoft DTP Plugin for Java にテスト結果をインポートできます（詳細についてはそれぞれの IDE 用の Parasoft DTP Plugin ユーザーズ ガイドを参照）。HTML レポートの構造の概要については、32 ページの「レポートの参照」を参照してください。

DTP Engine が DTP Server に接続されている場合、サーバーにもレポートが送信されます（38 ページの「Development Testing Platform (DTP) Server への結果の送信」を参照）。

レポートに表示されるファイルパスは、リソース パターンの構文に従います。15 ページの「リソース パターンの構文」を参照してください。

## レポートの出力場所の指定

解析時にレポートの出力ディレクトリを指定するには、`-report` スイッチを使用します。

```
-report location
```

### Ant および Maven のパターン

```
<report>location</report>
```

### プロパティ設定のパターン

```
report.location=location
```

`report.location` プロパティを使用して HTML レポートの場所を変更することもできます。

```
report.location=<HTML_REPORT_LOCATION>
```

## レポートのフォーマットの指定

`report.format` プロパティを設定すると、指定されたディレクトリに PDF レポートやカスタム拡張を使用したレポートを生成できます。詳細については 71 ページの「レポート設定」を参照してください。

```
report.format=pdf
```

## レポートの参照

作業ディレクトリまたは `report` スイッチで指定した場所に保存された `report.html` または `report.pdf` を開きます。解析のタイプによって、レポートに含まれるセクションは異なる可能性があります。以下のセクションは、すべての静的解析およびフロー解析コンフィギュレーションに含まれます。

## ヘッダー

### Jtest DTP Engine Report

DTP Engine for Java 10.2.3.201605130937

---

#### Session Summary

Build ID:	Jtest-2016-05-13
Test Configuration:	builtin://Demo Configuration
Started:	2016-05-13T12:14:19+02:00
Performed on:	jade by annstu
Session Tag:	`\${scontrol_branch}`-win32_x86_64
Project:	Jtest

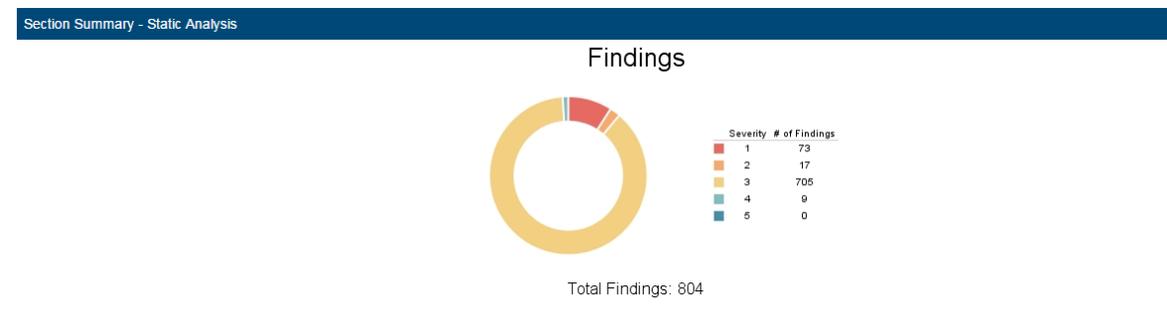
[Static Analysis](#) Severity 1 Findings: 73  
[Test Execution](#) Test Failures: 2/33

以下の情報が含まれています。

- 解析に使用されたツール
- ビルド ID
- テスト コンフィギュレーション
- 解析のタイムスタンプ
- マシン名とユーザー名
- セッション タグ
- プロジェクト名
- 重要度が最も高い指摘事項の数
- 失敗したテストの数

## Static Analysis

レポートの最初の部分は 2 つのセクションに分かれており、静的解析の指摘事項を表示します。1 番目のセクションには、指摘事項の概要が円グラフで示されます。グラフの色は重要度の種類を表し、対応する数字は静的解析で検出された指摘事項の数を表します。



2 番目のセクションには、静的解析の指摘事項の詳細が表示されます。このセクションは静的解析の結果を含む表から始まります。

▼ Details - Static Analysis

### Static Analysis

Module	Findings			Files		Lines	
	suppressed	total	per 10,000 lines	checked	total	checked	total
com.parasoft.demo	1	804	2213	59	59	3632	3632
<b>Total [0.00.43]</b>	<b>1</b>	<b>804</b>	<b>2213</b>	<b>59</b>	<b>59</b>	<b>3632</b>	<b>3632</b>

以下の情報が含まれています。

- モジュールの名前
- 抑制されたルールの数
- 指摘事項の総数
- 10,000 行あたりの指摘事項の平均数
- 解析されたファイルの数
- モジュール中のファイルの総数
- 解析されたコード行数
- モジュール中のコード行の総数

## All Findings

[All Findings] セクションには、指摘事項の詳細がカテゴリ別または重要度別に表示されます。表示を切り替えるには [重要度] または [カテゴリ] リンクをクリックします。

カテゴリ別の表示では、指摘事項はルールごとにカテゴリによって分類されています。解析スコープ内に各ルールの違反がいくつあるかも表示されます。

### All Findings by Category

[Category](#) | [Severity](#)

- [4] **Collections** (BD.CO)
- [4] Do not modify collection while iterating over it (BD.CO.ITMOD-1)
- [12] **Exceptions** (BD.EXCEPT)
- [12] Avoid NullPointerException (BD.EXCEPT.NP-1)
- [5] **Optimization** (BD.OPT)
- [3] Avoid inefficient removal of Collection elements (BD.OPT.INEFCOL-3)
- [1] Avoid inefficient iteration over Map entries (BD.OPT.INEFMAP-3)
- [1] Avoid inefficient removal of Map entries (BD.OPT.INEFMAPRM-3)

重要度ビューでは、指摘事項は重要度別に分類されています。重要度ごとの指摘事項の数も表示されます。

## All Findings by Severity

Category | Severity

- [73] Severity 1 - Highest
- [4] Do not modify collection while iterating over it (BD.CO.ITMOD-1)
  - [12] Avoid NullPointerException (BD.EXCEPT.NP-1)
  - [2] Avoid use before explicit initialization (BD.PB.NOTEXPLINIT-1)
  - [1] Avoid use of fields before initialization in constructors and static initializers (BD.PB.NOTINITCTOR-1)
  - [1] Do not append null value to strings (BD.PB.STRNULL-1)
  - [1] Avoid division by zero (BD.PB.ZERO-1)
  - [1] Protect against Command injection (BD.SECURITY.TDCMD-1)
  - [1] Protect against Environment injection (BD.SECURITY.TDENV-1)
  - [1] Protect against File contents injection (BD.SECURITY.TDFILES-1)
  - [1] Protect against File names injection (BD.SECURITY.TDFNAMES-1)
  - [1] Protect against Library injection (BD.SECURITY.TDLIB-1)
  - [1] Protect against Reflection injection (BD.SECURITY.TDRFL-1)
  - [2] Protect against SQL injection (BD.SECURITY.TDSQL-1)
  - [4] Protect against XML data injection (BD.SECURITY.TDXML-1)
  - [1] Protect against XSS vulnerabilities (BD.SECURITY.TDXSS-1)
  - [1] Ensure index is valid in JDBC method invocation (JDBC.BRSA-1)
  - [1] Use "PreparedStatement" correctly (JDBC.UFSC-1)
  - [6] Unrestricted lock resource (PB.CLOSE-1)
  - [1] Avoid conditional expressions that always evaluate to a constant value (PB.USC.CC-1)
  - [2] Avoid unreachable "else if" and "else" cases (PB.USC.UIF-1)
  - [18] Do not pass exception messages into output in order to prevent the application from leaking sensitive information (SECURITY.ESD.PEO-1)
  - [4] Avoid using "SELECT\*" in SQL queries (SECURITY.IBA.AUSS-1)
  - [4] Use "prepareCall" or "prepareStatement" instead of "createStatement" (SECURITY.IBA.UPS-1)
  - [1] Avoid passing hardcoded usernames/passwords/URLs to database connection methods (SECURITY.WSC.HCCS-1)
  - [1] Do not define instance fields in Servlet classes (SERVLET.IF-1)
- [17] Severity 2 - High
- [6] Avoid conditions that always evaluate to the same value (BD.PB.CC-2)
  - [2] Do not use "==" or "!=" to compare objects (PB.CUB.UEIC-2)
  - [1] Avoid using "private" fields which are never given a meaningful value (PB.USC.FCBS-2)
  - [4] Avoid "main()" methods because they may allow unauthorized access to classes (SECURITY.WSC.UMAIN-2)
  - [4] Use a Context Object to manage HTTP request parameters (SERVLET.UCO-2)
- [705] Severity 3 - Medium
- [3] Avoid inefficient removal of Collection elements (BD.OPT.INFCOL-3)
  - [1] Avoid inefficient iteration over Map entries (BD.OPT.INFMAP-3)

PDF 形式のレポートでは、これらのセクションがまとめて表示されます。

- [1] Erratic Application Behavior (SECURITY.EAB)
  - [1] Do not store user-given mutable objects directly into variables (SECURITY.EAB.SMO-3)
- [21] Serialization (SERIAL)
  - [21] Create a 'serialVersionUID' for all 'Serializable' classes (SERIAL.DUID-3)
- [6] Servlets (SERVLET)
  - [1] Do not define instance fields in Servlet classes (SERVLET.IF-1)
  - [4] Use a Context Object to manage HTTP request parameters (SERVLET.UCO-2)
  - [1] Do not use JDBC code in Servlet classes (SERVLET.AJDBC-3)
- [25] Unused Code (UC)
  - [10] Avoid unnecessary modifiers in an "interface" (UC.AAI-3)
  - [4] Remove commented out Java code (UC.ACC-3)
  - [11] Avoid local variables that are never read (UC.AURV-3)

Findings by Category section

- [73] Severity 1 - Highest
- [4] Do not modify collection while iterating over it (BD.CO.ITMOD-1)
  - [12] Avoid NullPointerException (BD.EXCEPT.NP-1)
  - [2] Avoid use before explicit initialization (BD.PB.NOTEXPLINIT-1)
  - [1] Avoid use of fields before initialization in constructors and static initializers (BD.PB.NOTINITCTOR-1)
  - [1] Do not append null value to strings (BD.PB.STRNULL-1)
  - [1] Avoid division by zero (BD.PB.ZERO-1)
  - [1] Protect against Command injection (BD.SECURITY.TDCMD-1)
- [17] Severity 2 - High
- [6] Avoid conditions that always evaluate to the same value (BD.PB.CC-2)
  - [2] Do not use "==" or "!=" to compare objects (PB.CUB.UEIC-2)
  - [1] Avoid using "private" fields which are never given a meaningful value (PB.USC.FCBS-2)
  - [4] Avoid "main()" methods because they may allow unauthorized access to classes (SECURITY.WSC.UMAIN-2)
  - [4] Use a Context Object to manage HTTP request parameters (SERVLET.UCO-2)

Findings by Severity section

## Findings by Author

このセクションには、解析されたコードに関する作成者と、作成者ごとの指摘事項の合計数が表示されます。作成者のリンクをクリックすると、指摘事項についての詳細情報を参照できます。

### Findings by Author

[Back to Top](#)

Author	Findings		
	suppressed	total	recommended
annstu	1	804	50
annstu Total Findings : 804			

[Back to Top](#)

/demo/src/examples/eval/Simple.java

8: Use 0 blank lines before the package statements

11: Use 1 blank line before type declaration

11: Use 1 blank line before every top-level class (or corresponding Javadoc)

FORMAT.U2BL-3

FORMAT.BLCD-3

FORMAT.U2BL-3

詳細ビューには、以下の情報が表示されます。

- 指摘事項があるファイルとその場所

- 違反メッセージとルール
- フロー解析レポートには、違反の原因 (C)、違反のポイント (P)、スローされた例外 (E) および重要なデータ フロー (!) がマーク付きで表示されます。

## Findings by File

[Findings by File] セクションでは、解析されたコードから、レポートされた指摘事項にアクセスすることができます。ノードの先頭の数字は、指摘事項の総数を表します。括弧内の数字は、そのノード中の抑制されたルールの数を表します。プラス記号 (+) の付いたノードをクリックすると、ノードを展開できます。PDF 形式のレポートでは、ノードはすべて展開された状態になっています。

### Findings by File

[Expand All](#) [Collapse All](#) [Back to Top](#)

+ 804 (1) Total (Suppressed)			
+ 804 (1) com.parasoft:demo			
+ 0 (0) META-INF			
+ 804 (1) src			
+ 804 (1) examples			
+ 10 (0) eval			
+ 10 (0) Simple.java			
8:	Use 0 blank lines before the package statements	annstu	FORMAT.U2BL-3
11:	Use 1 blank line before type declaration	annstu	FORMAT.BLCD-3
11:	Use 1 blank line before every top-level class (or corresponding Javadoc)	annstu	FORMAT.U2BL-3
19:	Use 1 blank line before every method declaration (or corresponding Javadoc)	annstu	FORMAT.U2BL-3
22:	This '@param' tag does not contain a meaningful description of the parameter	annstu	JAVADOC.MDJT-3
29:	Avoid label statement: 'case 10'	annstu	CODSTA.BPAULS-3
69:	"for" statement without a "{}" block	annstu	CODSTA.BPBLK-3
69:	Parameter 'match' dereferenced before being checked for null	annstu	DBC.IPAN-3
70:	"if" statement without a "{}" block	annstu	CODSTA.BPBLK-3
70:	Parameter 'str' dereferenced before being checked for null	annstu	DBC.IPAN-3
+ 396 (0) flowanalysis			
+ 88 (0) junit4			
+ 9 (0) metrics			
+ 97 (0) nbank			
+ 18 (0) queue			
+ 19 (0) security			
+ 14 (0) servlets			
+ 128 (0) stackmachine			
+ 17 (0) stubs			
+ 8 (1) suppress			
+ 0 (0) tests			

Flow Analysis Legend:

C - violation cause	E - throws an exception
P - violation point	! - important data flow

## メトリクス サマリー

実行したテスト コンフィギュレーションにメトリクス解析が含まれていた場合、レポートに [メトリクス] セクションが表示されます。詳細については 29 ページの「メトリクス 解析」を参照してください。

### Metrics Summary

[Expand All](#) [Collapse All](#) [Back to Top](#)

Metric name	Number of Items	Average	Std. Deviation	Maximum	Minimum
+ McCabe Cyclomatic Complexity (METRIC.CC) com.parasoft:demo	268	1,646	1,254	10	1
+ Nested Blocks Depth (METRIC.NBD) com.parasoft:demo	268	0,377	0,794	8	0
+ Number of Physical Lines in Files (METRIC.NOPLIF) com.parasoft:demo	58	62,552	65,087	460	11
+ Number of Source Lines in Methods (METRIC.NOSLIM) com.parasoft:demo	268	7,571	5,907	40	1

## Test Execution

レポートの 2 番目の部分は 2 つのセクションに分かれており、テスト実行の結果を表示します。1 番目のセクションには、テストの失敗とカバレッジの概要が円グラフで示されます。



2 番目のセクションには、テスト実行の詳細が表示されます。このセクションは、テスト実行の結果とカバレッジ情報を含む表から開始します。

Details - Test Execution

### Test Execution [Back to Top](#)

Module	Findings			Executed Test Cases				Coverage (%)
	fix unit test problems	review exceptions	review assertion failures	passed	failed	incompleted	total	line
com.parasoft.demo	2	0	0	31	2	0	33	13
Total [0:00:00]	2	0	0	31	2	0	33	13

以下の情報が含まれています。

- モジュール名
- 修正が必要な単体テストの問題の数
- レビューが必要な例外の数
- レビューが必要なアサーションの失敗の数
- 正常に実行された単体テストの数
- 単体テストの失敗の数
- 未完了の単体テストの数
- 単体テストの総数
- パーセンテージで表された行カバレッジ

## All Findings

[All Findings] セクションには、テストの実行中に検出された、すべての単体テストの問題の詳細情報が表示されます。

### All Findings

[2] Unit Test Problems  
[2] Assertion Failures  
[1] junit.framework.AssertionFailedError  
[1] java.lang.AssertionError

## Findings by Author

このセクションには、解析対象コードに関わる作成者の表と、作成者別の指摘事項の総数が表示されます。作成者のリンクをクリックすると、指摘事項の詳細情報を参照できます。

### Findings by Author

[Back to Top](#)

Author	Findings	
	total	recommended
<a href="#">annstu</a>	2	0

[annstu](#) Total Findings : 2 [Back to Top](#)

```
/demo/tests/examples/stubs/InterpreterTest.java [ 1 ]
Test case: testAdd34
value is:-1 [failure]
org.junit.Assert.fail(Assert.java:88)
org.junit.Assert.assertTrue(Assert.java:41)
examples.stubs.InterpreterTest.testAdd34(InterpreterTest.java:43)
```

詳細情報には、以下の項目が含まれます。

- 指摘事項の場所
- テスト名
- 失敗メッセージ

## Executed Tests (Details)

[Executed Tests (Details)] セクションでは、指摘事項を参照できます。すべてのテストが成功したノードは [P] で示されます。テストの失敗があったノードの場合、角括弧に数字が示されます。1 番目の値は、成功したテストの数です。2 番目の値は、そのノード中のテスト実行の総数です。[F] は、テストが失敗した最後のノードを表します。プラス記号 (+) の付いたノードをクリックすると、ノードを展開できます。

### Executed Tests (Details)

[Expand All](#) [Collapse All](#) [Back to Top](#)

```
+ [31/33] [0:00:00.794] Passed / Total
+ [31/33] [0:00:00.794] com.parasoft.demo
+ [P] [0:00:00.031] examples.junit4.MoneyTest
- [2/3] [0:00:00.025] examples.nbank.AccountTest
  [P] [0:00:00.008] testApply
  [P] [0:00:00.016] testApply2
- [F] [0:00:00.001] testApply3
```

## カバレッジ

このセクションは、テストの実行中に収集されたカバレッジの詳細を示します。各ノードの先頭には一連の値が表示されます。1 番目の値はカバレッジ率 (%) です。2 番目の値は、そのノード中で、テスト実行時にカバーされた行数です。3 番目の値は、ノード中の行数の合計です。プラス記号 (+) の付いたノードをクリックすると、ノードを展開できます。

### Coverage

[Expand All](#) [Collapse All](#) [Back to Top](#)

```
+ Total [13% 174/1315 executable lines]
+ com.parasoft.demo [13% 174/1315 executable lines]
+ src.examples [13% 174/1315 executable lines]
+ eval [0% 0/10 executable lines]
+ Simple.java [0% 0/10 executable lines]
+ flowanalysis [0% 0/489 executable lines]
+ junit4 [91% 91/100 executable lines]
+ Money.java [100% 30/30 executable lines]
+ MoneyBag.java [87% 61/70 executable lines]
```

## テストパラメーター

[ テストパラメーター ] セクションには、解析時に指定された引数が表示されます。

### Test Parameters

```
!jstcl -data C:\test-server\test\examples\demo\target\test\data.json -config builtin:/Demo Configuration -exclude path:C:\test-server\test\examples\demo\tests/** -settings C:\test-server\test\examples\demo\demo.properties -report C:\test-server\test\examples\demo\target\test -fail false
```

## Development Testing Platform (DTP) Server への結果の送信

DTP Server への接続の構成については 4 ページの「DTP Server への接続」を参照してください。テスト結果を DTP Server に送信するには、`-publish` スイッチを使用します。

```
-publish
```

### Ant および Maven のパターン

```
<publish>true</publish>
```

### プロパティ設定のパターン

```
report.dtp.publish=true
```

## DTP Server へのソースコードのパブリッシュ

デフォルトでは、レポートの設定が有効な場合、テスト対象ソースが DTP に送信されます。これにより、DTP は指摘事項に関連するソースコードを表示できるようになります。

`report.dtp.publish.src` 設定を使用すると、ソースコードのパブリッシュを無効にしたり、パブリッシュの深さを制限したり、DTP Server へのレポート送信が無効な場合にソースコードのパブリッシュを有効化したりできます。DTP Engine の設定に関するその他の情報については 62 ページの「設定リファレンス」を参照してください。

`report.dtp.publish.src` 設定には、以下のいずれかの値を指定できます。

- `off`: コードは DTP Server にパブリッシュされません。
- `min`: ソースの最小限の部分だけをパブリッシュします。ソース管理システムへの参照がないソースコードだけがパブリッシュされます。
- `full`: 指定されたスコープに関連するすべてのソースをパブリッシュします。これはデフォルトの設定値です。

DTP でのソースコードの参照については『Development Testing Platform ユーザーズガイド』を参照してください。

### コード解析を実行せずに DTP にソースをパブリッシュする

DTP Server にデータを送信するには、DTP Engine で解析を実行する必要がありますが、解析を実行せずにソースを送信することもできます。

1. 空のテスト コンフィギュレーションを作成し、`[INSTALL_DIR]/configs/user` に保存します (11 ページの「テスト コンフィギュレーションの指定」を参照)。
2. `report.dtp.publish.src` を適切に設定し、このコンフィギュレーションを実行します。

# Unit Test Connector

Unit Test Connector (UTC) を使用すると、オープンソースの単体テストツールで作成した単体テストを実行し、結果を DTP にレポートできます。現行の UTC for Java は、カスタマイズなしで次の単体テストツールをサポートします。

- JUnit 3 および 4
- TestNG

その他の単体テストツールとの統合については、Parasoft Marketplace (<http://marketplace.parasoft.com>) を確認してください。

## フレームワークごとのサポートの詳細

次の表は、各テストフレームワークに対するサポートの詳細です。

フレームワーク	サポート	未サポート
JUnit 3	<code>junit.framework.TestCase</code> を直接的に拡張し、 <code>test*</code> メソッドを持つテストクラス。	スイートクラス： <code>junit.framework.TestCase</code> を直接的に拡張し、 <code>static suite()</code> メソッドを持つクラス。テスト名は問わない。 <code>junit.framework.TestCase</code> を直接的に拡張し、 <code>runTest()</code> メソッドを持つテストクラス。
JUnit 4	<code>@org.junit.Test</code> アノテーションを持つテストメソッド。	<code>@org.junit.runners.Parameterized.Parameters</code> でパラメータライズされたテスト。 <code>@org.junit.experimental.theories.Theory</code> を持つセオリー。
TestNG	<code>@org.testng.annotations.Test</code> アノテーションを持つテストメソッド。 XML レポート (JUnit フォーマットのみ) を処理できます。	<code>@org.testng.annotations.DataProvider</code> でパラメータライズされたテスト。 <code>@org.testng.annotations.Test</code> アノテーションを持つクラス。 <code>@org.testng.annotations.Factory</code> でパラメータライズされたテスト。 TestNG フォーマットの XML レポートは処理できません。

## JUnit テストの実行

JUnit のテストは Maven、Ant、または Gradle フレームワークによって実行されます。テスト結果が収集され、各フレームワーク専用のプラグインによってレポートされます。Maven、Ant、または Gradle に関する問題のトラブルシューティングなど、プラグインの使用の詳細については、[manuals/ja/plugins-manual/index.html](http://manuals/ja/plugins-manual/index.html) を参照してください。

## 個々のテスト実行のタグ付け

特定のテスト実行から結果に割り当てることができるタグを定義するには、`session.tag` プロパティを使用します。このタグは、特定のモジュールに対する解析プロセスのユニークな識別子です。同じセッションタグを使用すると、前回の結果を表すデータを上書きできます。デフォルトでは、実行されたテスト コンフィギュレーション名がタグに設定されます。

```
session.tag=[name]
```

## テストと開発成果物との関連付け

要求、欠陥、タスク、フィーチャー リクエストといった多様な開発成果物とテストを関連付けるよう DTP Engine を設定できます。

単体テストと成果物を関連付けるには、次の操作を行う必要があります。

1. 成果物の関連付けを有効化します。
2. 課題トラッキング タグを指定し、URL の関連付けを設定します。
3. Javadoc でタグを使用します。

詳細については以下のセクションを参照してください。

### 成果物との関連付けの有効化

`report.associations` プロパティに `true` を設定すると、成果物との関連付けが有効になります。また、このプロパティは HTML レポートに表示されるテストとの関連付けも有効 / 無効にします。

```
report.associations=true
```

### 課題トラッキング タグの指定

レポートの関連付けが有効化された場合、デフォルトでは、各成果物のタイプに対して次のタグが関連付けられます。

- pr (欠陥)
- fr (エンハンスメント)
- task
- asset
- req (ユーザー ストーリー)

`issue.tracking.tags` プロパティを使用すると、その他のトラッキング タグを定義できます。タグ名はカンマで区切ります。

```
issue.tracking.tags=[tag1, tag2, . . .]
```

### 課題トラッキング タグおよび URL 関連付けの設定

HTML レポートに関連成果物へのリンクを生成できます。

```
report.assoc.url.[tag]=[URL]
```

URL には [%ID%] または \${id} 変数を含めることができます。変数は課題 ID に置換されます。  
例：

```
report.assoc.url.[tag]=http://bugzilla.company.com/show_bug.cgi?id=[%ID%]
```

## テスト詳細の有効化

HTML レポートにテストの詳細を表示するかどうかを指定できます。

```
report.contexts_details=[true | false]
```

テスト詳細の表示を有効にするには、`report.contexts_details` プロパティを `true` に設定する必要があります。

製品のプロパティ ファイルは、あらかじめテスト詳細を表示するよう設定されています。

詳細については 71 ページの「レポート設定」を参照してください。

## Javadoc タグの使用

Javadoc タグを使用して単体テストと成果物を関連付けることができます。Javadoc でタグを使用してテストと関連付けてください。Javadoc で使用するタグは、先頭に @ 文字を付けなければなりません。

```
/**
 * @bug 12345
 * @pr 223344
 * @tag1 5533
 */
@Test
public void testSomething()
{
    ...
}
```

タグとクラスを関連付けることもできます。クラスと関連付けると、そのクラスにあるすべてのテストがタグに関連付けられます。以下の例では、タグ 9876 に Test クラスの両方のテストが関連付けられます。一方、タグ 111 には testSomething2 テストだけが関連付けられます。

```
/**
 * @tag 9876
 */
public class Test {

    testSomething1()
    {
        ...
    }

    /**
     * @tag 111
     */
    testSomething2()
    {
        ...
    }
}
```

## 複数の関連付け

1 個のタグに複数の成果物を関連付けることができます。その場合、カンマで区切って指定します。カンマの後にスペースを入れてはいけません。

```
/**
 * @task 1234,2345
 */
@Test
public void testSomething()
{
    ...
}
```

カンマの後にスペースを入れて区切った場合、1 番目の成果物だけがテストに関連付けられます。以下の例では、テストはタスク 1234 にだけ関連付けられます。

```
/**
 * @task 1234, 2345
 */
@Test
public void testSomething()
{
    ...
}
```

## カバレッジの収集

JUnit テストの実行時に、テストのカバレッジが収集されます。コード カバレッジの収集および DTP へのレポートについては、45 ページの「Code Coverage Engine」を参照してください。

# Code Coverage Engine

セクションの内容：

- 単体テストのカバレッジ
- アプリケーション カバレッジ

# 単体テストのカバレッジ

以下のビルド システムで特別なコマンドを使用すると、単体テスト実行時にカバレッジ情報を収集できます。

- Maven
- Ant
- Gradle

単体テスト実行中に収集されたカバレッジ情報の関連付けは、JUnit 4 でだけサポートされています。

ビルド システム プラグインの詳細な設定方法や、テストを実行して実行時カバレッジを収集する方法については、次のビルドシステム プラグイン マニュアルを参照してください。

[INSTALL]/manuals/ja/plugins-manual/index.html

カバレッジ情報を収集して DTP に送信するには、Unit Tests ビルトイン コンフィギュレーションを使用してテストを実行します。次のコマンドは Maven を使用する場合の例です。

```
mvn clean jtest:instrument jtest:jtest -Djtest.config="builtin://Unit Tests"
```

単体テストのセットアップと実行については、40 ページの「Unit Test Connector」を参照してください。

## カバレッジ データのマージ

DTP でカバレッジ データを適切にマージするには、コマンド ラインまたは .properties 設定ファイルで 1 つ以上のカバレッジ イメージ タグを指定する必要があります。カバレッジ イメージは接続された DTP に自動的に送信され、DTP でフィルターとカバレッジ イメージを関連付けることができます。

report.coverage.images プロパティを使用すると、DTP Server でカバレッジ イメージを作成するのに使用できるタグを 3 つまで指定できます。

```
report.coverage.images=[tag1; tag2; tag3]
```

DTP の Report Center 管理ページ ([ 管理 ] > [ プロジェクト ] > フィルターをクリック ) でカバレッジ イメージを関連付けます。

また、report.coverage.limit プロパティを使用すると、カバレッジの下限を指定することもできます。

```
report.coverage.limit=[value]
```

この値よりも低いカバレッジは、レポートで強調表示されます。デフォルト値は 40 です。

# アプリケーション カバレッジ

実行中の Web アプリケーション サーバーで手動または自動での機能テストを実行し、カバレッジ データをモニターおよび収集できます。カバレッジ データおよびテスト結果を DTP サーバーにレポートし、データをマージして関連付けることもできます。アプリケーション カバレッジ情報を DTP のカバレッジ エクスプローラー（詳細は DTP ユーザーズ ガイドの「カバレッジ エクスプローラー」を参照）に表示し、アプリケーションがどの程度テストされたかや、テストの品質を知ることができます。

## 前提条件

カバレッジを収集するには、以下のコンポーネントが必要です。

- Java JDK 1.5
- Apache Maven、Gradle、または Ant ビルド システム

Java JDK 8 でのバイトコードの改行の計算方法は、それより前のバージョンの Java と異なります。そのため、カバレッジの結果が変わる場合があります。

## プロセスの概要

DTP Engine for Java には、カバレッジ エージェントというコンポーネントが付属しています。カバレッジ エージェントは、テスト対象アプリケーション (AUT) にアタッチされ、AUT の実行されたコードをモニターします。AUT にカバレッジ エージェントをアタッチすると、REST API が公開され、テストおよびテスト セッションの開始および終了をマークできるようになります。

カバーされるコード行に関するメタデータ（静的カバレッジ データ）は、アプリケーションのビルド プロセスの一環として特別なテスト コンフィギュレーションを実行することによって収集されます。テスト実行時には、カバレッジ エージェントとのやりとりが動的カバレッジ マップに書き込まれます。このマップには、実行されたコード行を示すマーカーが記録されます。

DTP Engine は、この動的カバレッジ マップと静的カバレッジ データを処理します。カバレッジ情報を保存した coverage.xml が生成され、DTP Server に送信されます。DTP Server は、カバレッジ データを受信すると、カバレッジ イメージにロードします。カバレッジ イメージは、同じビルド ID を持つ複数の実行のカバレッジ データを集約することを可能にする特別なタグです。カバレッジ イメージにより、カバレッジ情報と特定のテストを関連付けることが可能になります。

テスト実行ツールの結果 (SOAtest のテスト、DTP Engine によって実行されたテスト、手動テストなど) も report.xml という形で DTP Server に送信されます。カバレッジ データ ファイルとレポートのビルド ID が一致すると、DTP Server はそれらのデータを関連付け、カバレッジ情報に表示します。

プロセスの詳細については、以下のセクションで説明します。

## カバレッジ収集のためのテスト対象アプリケーション (AUT) の構成

AUT の準備には、以下の手順が含まれます。

1. 静的カバレッジ ファイルを生成します。静的カバレッジ ファイルには、ユーザー クラス、メソッド、行などの情報が記録されます。詳細は 48 ページの「静的カバレッジ ファイルの生成」で説明されています。
2. AUT にカバレッジ エージェントをアタッチします。49 ページの「テスト対象アプリケーション (AUT) へのカバレッジ エージェントのアタッチ」を参照してください。
3. カバレッジ エージェントには、ファイルを出力したり、スコープを決定するためのデフォルトの設定が含まれていますが、アプリケーション カバレッジの目的に合わせて構成ファイルのプロパティを設定できます。詳細については 49 ページの「カバレッジ エージェントの構成」を参照してください。

## 静的カバレッジ ファイルの生成

静的カバレッジ ファイルは、ビルド プロセス中に Jtest maven プラグインによって生成されます。このファイルは、ソース コードがあるビルド マシン上で生成される必要があります。生成された静的カバレッジ ファイルは、コードが変更されるまで使用できます。

静的カバレッジ ファイルを生成するには、AUT のメイン ディレクトリで、次のコマンドを実行します。

Maven の場合：

```
mvn jtest:monitor
```

Gradle の場合：

```
gradle assemble jtest-monitor -I [INSTALL]/integration/gradle/init.gradle
```

Ant は、モニター タスクを実行する前にすべてのクラスをコンパイルすることを要求します。ビルドの前にプロジェクトを変更し、正しい順序になるようタスクを設定してください。以下はターゲットの設定例です。

```
<target name="jtest-monitor" depends="compile">
    <jtest:monitor/>
</target>
```

以下のコマンドを AUT のメイン ディレクトリで実行し、プロジェクトをビルドして静的カバレッジ ファイルを含むパッケージを生成します。

```
ant -lib [INSTALL]/integration/ant/jtest-ant-plugin.jar -listener
com.parasoft.Listener jtest-monitor
```

monitor.zip パッケージが生成され、ビルド出力ディレクトリに置かれます。その場所へのパスがコンソールに出力されます。

このパッケージには以下が含まれます。

- static\_coverage.xml - 静的カバレッジ情報が保存されています。
- agent.jar - Jtest Java カバレッジ エージェントの jar アーカイブです。

- agent.properties - エージェントの設定ファイルです。ビルド プロセス実行中に生成されたスコープ パラメーターや、他の属性が保存されています。
- agent.sh/agent.bat - エージェントを AUT プロセスにアタッチするために必要な Jtest Java エージェントの VM 引数を生成するスクリプトです。

## テスト対象アプリケーション (AUT) へのカバレッジ エージェントのアタッチ

monitor.zip パッケージの内容をサーバー マシンに展開し、agent.sh/agent.bat スクリプトを実行して Jtest Java エージェントの VM 引数を生成します。

monitor.zip パッケージのスクリプトは、コンソールに `-javaagent` VM 引数を出力します。エージェントを AUT にアタッチするには、この引数が必要です。結果は `runtime_coverage` サブ ディレクトリに出力されます。

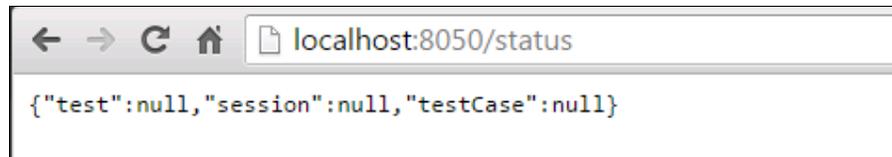
```
Jtest Agent VM argument:
-javaagent:"[path to agent dir]\agent.jar"=settings="[path to agent properties
file]\agent.properties",runtimeData="[path to monitor dir]\monitor\
runtime_coverage"
```

アプリケーション サーバーの起動スクリプトに `-javaagent` フラグを追加してサーバーを再起動します。具体的な手順については、チュートリアルで、55 ページの「ステップ 5a: Apache Tomcat からの実行時データの収集」、56 ページの「ステップ 5b: JBoss および WildFly からの実行時データの収集」、および 58 ページの「ステップ 5c: Oracle WebLogic Server からの実行時データの収集」を参照してください。

AUT にカバレッジ エージェントをアタッチすると、エージェントを制御するための REST API が公開されます。ブラウザに次の URL を入力して、カバレッジ エージェントが設定されていることを確認します。

[application\_host]:8050/status/

アプリケーションが適切に構成されている場合、API は次のデータを返します。



## カバレッジ エージェントの構成

通常、アプリケーション サーバーには複数のアプリケーションがあります。さらに、共通サーバー クラスまたはアプリケーション ライブラリはインストールする必要がありません。Jtest DTP Engine がカバレッジを収集する必要があるのは、アプリケーションのソース コードだけです。すべてのクラスをインストールすると、時間がかかりすぎる場合があります。

サーバー上のアプリケーションは既にビルドされており、ソース コードからどのクラスが生成されたかに関する情報を収集することはできません。このため、Jtest エージェントのスコープを適切に設定することが非常に重要です。

カバレッジ エージェントは、agent.properties ファイルまたは `-javaagent` への引数で設定できます。agent.properties ファイルは monitor.zip パッケージに生成されます (48 ページの「静的カバレッジ ファイルの生成」を参照)。このファイルでプロパティを変更してカバレッジ エン

トを適切に設定できます。以下は、agent.properties ファイルを使ってカバレッジ エージェントを設定する例です。

```
jtest.agent.runtimeData=[path to runtime_coverage directory]
jtest.agent.includes=com/myapp/data,com/myapp/common/**
jtest.agent.excludes=com/myapp/transport/*,com/myapp/autogen/**
jtest.agent.autostart=false
```

-javaagent にプロパティを追加するには、jtest.agent 接頭辞を削除して名前を変更する必要があります。

```
-javaagent:"[path to agent dir]\agent.jar"=settings="[path to agent properties file]\agent.properties",autostart=true
```

-javaagent への引数として渡されたプロパティは、agent.properties ファイルで設定されたプロパティを上書きします。上記の例の場合、autostart=true は jtest.agent.autostart=false を上書きします。

以下の表は、カバレッジ エージェントを設定するために使用できる全プロパティの一覧です。

プロパティ	説明
jtest.agent.runtimeData	<p>実行時データが保存される場所です。次の例では、C:/tmp/myapp/ ディレクトリに runtime_ という接頭辞の付いたファイルが作成されます。</p> <p>'C:/tmp/myapp/runtime'</p>
jtest.agent.includes	<p>インストゥルメントするクラスのパターンをカンマ区切りのリストで指定します。次のワイルドカードがサポートされています。</p> <p>* はゼロ個以上の文字に一致します。 ** は複数のディレクトリ レベルに一致します。</p> <p>次の例では、com.myapp.data パッケージのすべてのクラスと com.myapp.common で始まるパッケージおよびサブパッケージのすべてのクラスがインストゥルメントされます。</p> <p>com/myapp/data/*,com/myapp/common/**</p>
jtest.agent.excludes	<p>インストゥルメントから除外するクラスのパターンをカンマ区切りのリストで指定します。次のワイルドカードがサポートされています。</p> <p>* はゼロ個以上の文字に一致します。 ** は複数のディレクトリ レベルに一致します。</p> <p>次の例では、com.myapp.transport パッケージのすべてのクラスと com.myapp.autogen で始まるパッケージおよびサブパッケージのすべてのクラスがインストゥルメントから除外されます。</p> <p>com/myapp/transport/*,com/myapp/autogen/**</p>

プロパティ	説明
<code>jtest.agent.autostart</code>	実行時データの自動収集を有効 / 無効にします。デフォルト値は <code>true</code> です。
<code>jtest.agent.port</code>	エージェントの通信ポートを設定します。デフォルト値は 8050 です。
<code>jtest.agent.debug</code>	コンソールへの冗長な出力を有効 / 無効にします。デフォルト値は <code>false</code> です。
<code>jtest.agent.collectTestCoverage</code>	テスト ケースのカバレッジ情報の収集を有効 / 無効にします。デフォルト値は <code>false</code> です。
<code>jtest.agent.enableMultiuserCoverage</code>	複数ユーザーの Web アプリケーション カバレッジの収集を有効 / 無効にします。デフォルト値は <code>false</code> です。  このプロパティを <code>true</code> に設定すると、Coverage Agent Manager を使って複数ユーザーのカバレッジを収集できます。詳細については Parasoft DTP ドキュメントの「Coverage Agent Manager (CAM)」のセクションを参照してください。

## テスト コンフィギュレーションと実行

SOAtest を使用して機能テストを実行したり (テスト コンフィギュレーションの設定については、『SOAtest ユーザーズ ガイド』の「SOAtest のアプリケーション カバレッジ」を参照してください)、手動テストを実行できます。テスト セッション終了時に、SOAtest で指定されたディレクトリの `runtime_coverage_[timestamp].data` ファイルにカバレッジが保存されます。この情報と静的カバレッジ データをマージして `coverage.xml` ファイルを作成し、DTP にアップロードすることができます。

## DTP への結果のアップロード

SOAtest を使用して実行したテストについては、SOAtest の XML レポートを DTP にアップロードする必要があります。詳細については、『SOAtest ユーザーズ ガイド』の「SOAtest のアプリケーション カバレッジ」セクションの「DTP へのテスト結果のアップロード」を参照してください。

## 動的カバレッジ データ ファイルの生成と DTP へのアップロード

適切にカバレッジ データをマージするには、`jtest.properties` ファイルで以下の設定を行う必要があります。

- `report.coverage.images` - DTP Server でカバレッジ イメージを作成するために使用するタグのセットを指定します。カバレッジ イメージは、同じビルド ID を持つ複数の実行のカバレッジ データを集約する際に一意の識別子となります。DTP は、1 レポートにつき最大 3 つまでのカバレッジ イメージをサポートします。
- `session.tag` - 同じビルドの個々のテスト実行を区別する一意の識別子を指定します。

- `build.id = "[value]"` - 結果にラベル付けするためのビルド識別子を指定します。ビルドごとに固有の値を使用することも、特定のビルドで実行される複数のテストセッションで同じ値を使用することもできます。

これらの設定に加えて、スコープ、作成者、DTP などに関する他のプロパティを指定する必要があります。詳細については以下のセクションを参照してください。

- 4 ページの「DTP Server への接続」
- 38 ページの「Development Testing Platform (DTP) Server への結果の送信」
- 62 ページの「設定リファレンス」

`coverage.xml` ファイルに実行時カバレッジ データを書き込むには、静的カバレッジ データだけでなく、テスト実行時に生成される実行時データ カバレッジにも DTP Engine がアクセスできる必要があります。次の引数で必要なデータを指定して Calculate Application Coverage テスト コンフィギュレーションを実行します。

```
jtestcli -staticcoverage [path to static_coverage.xml file] -runtimecoverage [path/dir] -config "builtin://Calculate Application Coverage"
```

## DTP でのカバレッジの参照

DTP のカバレッジ エクスプローラーを使用して、テスト実行時のアプリケーション カバレッジを参照できます。カバレッジ情報の参照については、DTP のマニュアルを参照してください。

## Web アプリケーション カバレッジのチュートリアル

次のチュートリアルでは、Jtest DTP Engine を使用してアプリケーション カバレッジを収集する手順を説明します。サーバーの実行やアプリケーションのデプロイなどの基本的なトピックも説明されているため、ビギナーにも上級ユーザーにも役に立ちます。

### 前提条件

次の前提条件は、このチュートリアルを実行する場合にだけ必要です。Web アプリケーション カバレッジは、Java Agent を利用できる任意のサーバーで収集できます。

- Java JDK 1.5
- Apache Maven ビルド システム
- Calculator サンプル。このサンプルは、非常に単純な Web アプリケーションです。サンプルは `[INSTALL]/examples` ディレクトリにあります。
- 以下のアプリケーション サーバーのいずれか
  - Apache Tomcat (バージョン 6.0、7.0、8.0 でこのシナリオを検証済み)
  - JBoss/WildFly
  - Oracle WebLogic

## ステップ 1: Calculator サンプルの準備

アプリケーションはサーバー上で WAR ファイル (Web Application Archive) にパッケージ化する必要があります。DTP Engine に付属の maven-war-plugin を使用してアプリケーションをパッケージ化します。プラグインは自動的に適切な WAR 構造を持ったアプリケーションをビルドします。アプリケーション ディレクトリで次のコマンドを実行します。

```
mvn clean install
```

プロジェクトのビルドが終了したら、target ディレクトリに WAR ファイルが配置されます。  
Calculator\target\Calculator.war

## ステップ 2a: Tomcat アプリケーション サーバーにサンプルをデプロイする

このステップを続行する前に、53 ページの「ステップ 1: Calculator サンプルの準備」の説明に従って WAR ファイルをビルドする必要があります。JBoss/WildFly アプリケーション サーバーを使用する場合は、54 ページの「ステップ 2b: WildFly/JBoss アプリケーション サーバーにサンプルをデプロイする」に進んでください。WebLogic アプリケーション サーバーを使用している場合は、55 ページの「Step 2c: Oracle WebLogic Server にサンプルをデプロイする」に進んでください。

アプリケーション サーバーへのデプロイにはさまざまな方法がありますが、このチュートリアルでは、Tomcat Web アプリケーションマネージャを使用してリモートからデプロイする方法を説明します。

1. [Apache Tomcat インストール ディレクトリ]/conf/tomcat-users.xml を開いて <tomcat-users> セクションに次の行を追加することで、新しいユーザーを追加します。

```
<user username="tomcat" password="tomcat" roles="tomcat, manager-gui"/>
```

2. 次のコマンドを実行してサーバーを実行します。

Windows

```
[Your apache-tomcat installation directory]/bin/startup.bat
```

Linux

```
[Your apache-tomcat installation directory]/bin/startup.sh
```

3. ブラウザーで次の URL を開き、ステップ 1 で設定したユーザー名およびパスワード (tomcat/tomcat) を入力します。

```
http://localhost:8080/manager/html
```

4. [WAR ファイルの配備] セクションで [ファイルを選択] をクリックし、Calculator.war を参照します。
5. [配備] をクリックします。次の URL でアプリケーションを使用できるようになります。

```
http://localhost:8080/calculator
```

6. アプリケーションを操作して正しくデプロイされたことを確認します。

## ステップ 2b: WildFly/JBoss アプリケーション サーバーにサンプルをデプロイする

アプリケーション サーバーへのデプロイにはさまざまな方法がありますが、このチュートリアルでは、JBoss Web コンソールを使用してデプロイする方法を説明します。この手順は JBoss AS 7.1.1、JBoss EAP 6.4、WildFly 8.2 および 9.0 サーバーに適用できます。

1. Windows で add-user スクリプトを実行し、新しいユーザーを追加します。

```
[JBoss installation directory]/bin/add-user.bat
```

2. ユーザーのタイプを選択するよう求められたら、Management User を選択します。

```
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a): a
```

3. 入力を求められたら、ユーザー名およびパスワードを入力し、Realm フィールドは空のままにします。

```
Enter the details of the new user to add.
Realm (ManagementRealm) :
Username : admin
Password : ladm-adm-adm
```

4. 次のスクリプトを実行してサーバーを実行します。

```
[JBoss installation directory]/bin/standalone.bat
```

5. ブラウザーで次の URL を開き、ステップ 3 で設定したユーザー名およびパスワード (admin/ladm-adm-adm) を入力します。

```
http://localhost:9990/console/App.html#deployments
```

6. [Add] をクリックし (JBoss 7.1.1 の場合は [Add Content])、Calculator.war ファイルを参照します。
7. [Next] をクリックし、[Save] をクリックします。
8. デプロイメント リストで Calculator.war を有効化し、ブラウザで次の URL を開きます。

```
http://localhost:8080/Calculator
```

9. アプリケーションを操作して正しくデプロイされたことを確認します。

## Step 2c: Oracle WebLogic Server にサンプルをデプロイする

WebLogic アプリケーション サーバーにデプロイする方法は色々ありますが、このチュートリアルでは管理コンソールを使ってリモートでデプロイする方法を説明します。この操作はバージョン 12.2.1 および 10.3.6 で検証済みです。

このチュートリアルは、WebLogic Server がインストール済みであること、およびサーバー ドメインが作成済みであることを前提とします。

1. 管理コンソールに移動します (<http://localhost:7001/console>)。
2. [ドメイン構成] パネル メニューで、[デプロイメント] をクリックして [デプロイメントのサマリー] セクションを開きます。
3. [デプロイメント] の表で [インストール] ボタンをクリックします。
4. [パス] フィールドで、WAR ファイルへのパスを指定します。
5. [次] をクリックします。
6. [このデプロイメントをアプリケーションとしてインストールする] オプションを選択します。
7. [次] をクリックします。
8. [終了] をクリックします。
9. アプリケーションとやり取りしてその機能をテストします (<http://localhost:7001/Calculator>)。

## ステップ 3: メタデータ ファイルの準備

サンプル アプリケーションのメイン ディレクトリで、次のコマンドを実行します。

```
mvn jtest:monitor
```

コンソールに出力された `-javaagent` フラグをコピーします。ゴールはアプリケーション カバレッジの収集に必要な成果物を含む `.zip` パッケージを生成します。詳細については 48 ページの「静的カバレッジ ファイルの生成」を参照してください。

## ステップ 4: javaagent VM 引数の生成

`monitor.zip` パッケージの内容をサーバー マシンに展開し、`agent.bat` (Windows) または `agent.sh` (Linux) スクリプトを実行します。`javaagent` フラグが生成され、コンソールに出力されます。

以下は `javaagent` フラグの出力例です。

```
-javaagent:"E:\Parasoft\JTest\examples\calculator\target\jtest\monitor\monitor\agent.jar"=settings="E:\Parasoft\JTest\examples\calculator\target\jtest\monitor\monitor\agent.properties",runtimeData="E:\Parasoft\JTest\examples\calculator\target\jtest\monitor\monitor\runtime_coverage"
```

## ステップ 5a: Apache Tomcat からの実行時データの収集

JBoss/WildFly サーバーを使用する場合、56 ページの「ステップ 5b: JBoss および WildFly からの実行時データの収集」に移動してください。

1. スクリプト ファイルを開きます。

Windows

```
[Your apache-tomcat installation directory]/bin/catalina.bat
```

Linux

```
[Your apache-tomcat installation directory]/bin/catalina.sh
```

2. スクリプトの先頭に javaagent フラグを追加します。

Windows

```
if "%1"=="stop" goto skip_instrumentation
set JAVA_OPTS=%JAVA_OPTS% [generated javaagent flag]
:skip_instrumentation
```

Linux

```
if [ "$1" = "start" -o "$1" = "run" ]; then
export JAVA_OPTS="$JAVA_OPTS [generated javaagent flag]"
fi
```

-javaagent フラグは 1 行で記述します。

3. サーバーを再起動し、ブラウザで次の URL を開きます。

```
http://localhost:8080/Calculator
```

4. アプリケーションを操作してサーバーを停止します。

Jtest Agent は、`mvn package jtest:monitor` ゴールが生成した `-javaagent` フラグの `runtimeData` プロパティに従って実行時データを書き込みます。デフォルトでは、実行時データは `[path to monitor dir]/monitor/runtime_coverage` ディレクトリに書き込まれます。

## ステップ 5b: JBoss および WildFly からの実行時データの収集

標準的な Java 階層クラス ローダーは、J2EE サーバーには適していません。なぜなら、JAR ファイルが使用されるかどうかにかかわらず、常にロードされるからです。アプリケーションは JAR ファイルを必要になったときにロードすることはできません。さらに、JAR ファイル間の可視性を制限するための適切なソリューションがありません。これによって、1 つのライブラリの 2 つのバージョンの間で競合が生じることがよくあります。また、すべてを含む巨大なクラス ローダーが作成される原因にもなります。

JBoss および WildFly サーバーは、これらの課題を解決するために JBoss モジュールを使用します。モジュール化された非階層的クラス ロードの実装は、JBoss OSGi および JBoss Java EE 実装の基本でもあります。

しかし、このクラス ローダーの優位性は、サーバーからの実行時データの収集という点では、新たな問題を引き起こします。Jtest Agent ライブラリはアプリケーション クラスから参照できなければなりませんが、JBoss モジュールによってそれが不可能になるのです。これが `-javaagent` フラ

グを追加する理由の 1 つです。このフラグを追加していない場合、`ClassNotFoundException` が返されます。

そのため、`jboss.modules.system.pkgs` システム プロパティも使用する必要があります。しかし、それにはまた別の問題もあります。JBoss は自身のスクリプト内でこのフラグを設定します。そのため JBoss と Jtest Agent の統合が複雑になります。これらのスクリプトが前に設定されたスクリプトを上書きする可能性があるからです。

## JBoss AS 7.1/JBoss EAP 6.4/WildFly 8.2/9.0

1. セットアップ スクリプトに `-javaagent` フラグを追加します。

Windows:

- a. `[Your jboss installation directory]/bin/standalone.conf.bat` を開き、ファイルの最後にある `JAVA_OPTS` に `javaagent` フラグを追加します。

```
set "JAVA_OPTS=%JAVA_OPTS% [generated javaagent flag]"
```

`" :JAVA_OPTS_SET"` はファイルの最後の行のままにします。

`-javaagent` フラグは 1 行で記述します。

Java 7 で動作するサーバーの場合、`JAVA_OPTS` に `-XX:-UseSplitVerifier` フラグを追加します。

Java 8 で動作するサーバーの場合、`JAVA_OPTS` に `-noverify` フラグを追加します。

- b. 次のセクションを変更して `boss.modules.system.pkgs` 設定に `javaagent` クラスを追加します。

```
set "JAVA_OPTS=%JAVA_OPTS% -Djboss.modules.system.pkgs=org.jboss.byteman"
```

変更後

```
set "JAVA_OPTS=%JAVA_OPTS% -
Djboss.modules.system.pkgs=org.jboss.byteman,com.parasoft.jtest.instrumentation,shaded.com.parasoft.jtest.runtime"
```

- c. `standalone.bat` スクリプトを使用してサーバーを再起動します。

Linux:

- a. `[Your jboss installation directory]/bin/standalone.conf` を開き、ファイルの最後にある `JAVA_OPTS` に `javaagent` フラグを追加します。

```
set "JAVA_OPTS=$JAVA_OPTS [generated javaagent flag]"
```

`-javaagent` フラグは 1 行で記述します。

Java 7 で動作するサーバーの場合、`JAVA_OPTS` に `-XX:-UseSplitVerifier` フラグを追加します。

Java 8 で動作するサーバーの場合、`JAVA_OPTS` に `-noverify` フラグを追加します。

- b. 次のセクションを変更して `boss.modules.system.pkgs` 設定に `javaagent` クラスを追加します。

```
JBOSS_MODULES_SYSTEM_PKGS="org.jboss.byteman"
```

#### 変更後

```
JBOSS_MODULES_SYSTEM_PKGS="org.jboss.byteman,com.parasoft.jtest.instrumentation,s
haded.com.parasoft.jtest.runtime"
```

- c. `standalone.sh` スクリプトを使用してサーバーを再起動します。

2. Web ブラウザーで次の URL を開きます。

```
http://localhost:8080/Calculator/
```

3. アプリケーションを操作してサーバーを停止します。

Jtest Agent は Jtest の target 内の maven ディレクトリ (`target/jtest`) に `runtimeData.data` ファイルを作成します。この場所は `.json` データ ファイルに書き込まれるため、場所を覚えておく必要はありません。

### JBOss 6.1/5.1

古いバージョンの JBOSS では、`jboss.modules.system.pkgs` プロパティを設定する必要はありません。

Windows:

1. `[Your jboss installation directory]/bin/run.conf.bat` を開き、ファイルの最後にある `JAVA_OPTS` に `javaagent` フラグを追加します。

```
set "JAVA_OPTS=%JAVA_OPTS% [generated javaagent flag]"
```

`":JAVA_OPTS_SET"` はファイルの最後の行のままにします。

`-javaagent` フラグは 1 行で記述します。

2. `standalone.bat` スクリプトを使用してサーバーを再起動します。

Linux:

1. `[Your jboss installation directory]/bin/run.conf.bat` を開き、ファイルの最後にある `JAVA_OPTS` に `javaagent` フラグを追加します。

```
set "JAVA_OPTS=$JAVA_OPTS [generated javaagent flag]"
```

`-javaagent` フラグは 1 行で記述します。

2. `run.sh` スクリプトを使用してサーバーを再起動します。

## ステップ 5c: Oracle WebLogic Server からの実行時データの収集

1. スクリプト ファイルを開きます。

## Windows

```
[Your weblogic installation directory]/user_projects/domains/[your domain]bin/
startWebLogic.cmd
```

## Linux

```
[Your weblogic installation directory]/user_projects/domains/[your domain]bin/
startWebLogic.sh
```

2. START WEBLOGIC コメントで始まるセクションで、JAVA\_OPTIONS に `-javaagent` フラグを追加します。`-javaagent` フラグは 1 行で記述しなければなりません。

## Windows

```
@REM START WEBLOGIC
set JAVA_OPTIONS=%JAVA_OPTIONS% [generated javaagent flag]
```

## Linux

```
# START WEBLOGIC
export JAVA_OPTIONS=$JAVA_OPTIONS [generated javaagent flag]
```

3. サーバーを再起動し、次の URL をブラウザで開きます。  
`http://localhost:7001/Calculator`
4. アプリケーションとやり取りします。
5. サーバーを停止します。

Jtest Agent は、`mvn package jtest:monitor` ゴールが生成した `-javaagent` フラグの `runtimeData` プロパティに従って実行時データを書き込みます。デフォルトでは、実行時データは `[path to monitor dir]/monitor/runtime_coverage` ディレクトリに書き込まれます。

## ステップ 6: カバレッジ レポートの生成

Calculate Application Coverage テスト コンフィギュレーションを実行します。`-staticcoverage` および `-runtimecoverage` パラメーターを使って、`static_coverage.xml` ファイルと実行時カバレッジデータのディレクトリを渡します。

```
jtestcli -staticcoverage [path to static_coverage.xml file] -runtimecoverage
[path/dir] -config "builtin://Calculate Application Coverage"
```

`static_coverage.xml` ファイルへのパスは、このファイルを含む `monitor.zip` パッケージの内容を展開したディレクトリでなければなりません。

`runtime_coverage` ディレクトリへのパスは、`-javaagent` VM 引数の `runtimeData` プロパティとしてコンソールに出力されます。

# DTP Engine for Java のカスタマイズ

デフォルトの設定は、[INSTALL\_DIR]にある `jtestcli.properties` ファイルに指定されています。`-settings` スイッチの後ろに設定ファイルへの絶対パスまたは相対パスを指定することで、Static Analysis Engine をカスタマイズできます。`-settings` スイッチを複数回使用して、複数のプロパティ ファイルを指定できます。

```
jtestcli -settings path/to/settings.properties
-jsettings path/to/another/settings.properties
```

`jtestcli.properties` ファイルの設定が最初に読み取られ、その他のファイルで指定された設定が `jtestcli.properties` ファイルの設定を上書きします。`$USER_HOME` ディレクトリの `jtestcli.properties` ファイルに共通設定を保存し、[INSTALL\_DIR] ディレクトリのプロパティ ファイルの設定を上書きすることもできます。

全般的な設定は次の順序で適用されます。

1. [INSTALL\_DIR]/etc/jtestcli.properties  
Static Analysis Engine 用の基本構成であり、変更してはいけません。
2. [INSTALL\_DIR]/jtestcli.properties  
共通に使用される設定のテンプレートがあります (ライセンス、レポートなど)
3. (任意) [USER\_HOME]/jtestcli.properties
4. (任意) [WORKING\_DIR]/jtestcli.properties
5. コマンドライン スイッチ `-settings path/to/settings.properties` で渡されたカスタム設定 (例: `-settings ../settings.properties`)
6. コマンドライン スイッチ `-property [key=value]` で渡されたカスタム設定

上記のすべての設定は、コマンドライン スイッチ (例: `- report`, `-config`, `-dtp.share.enabled`) から渡されるカスタム設定で上書きすることができます。

## Ant および Maven の設定階層

```
<settings>path/to/settings.properties</settings>
```

### Maven の設定階層

Maven を使用している場合、以下によって `jtestcli` の設定を行うことができます。

- `-Djtest.[Maven property name]`  
(例: `-Djtest.settings="my.general.properties"`)
- `pom.xml` ファイル
- `-Dproperty.jtest.[property name]`  
(例: `-Dproperty.jtest.license.use_network=true`)

`-Djtest` と `pom.xml` は、どちらも `-Dproperty.jtest` で渡された設定を上書きします。

`-Djtest` は、`pom.xml` の設定がユーザー プロパティとして提供されている場合、`pom.xml` の設定を上書きします。しかし、設定の値が直接 `pom.xml` で指定されている場合は、この値が最優先され、コマンドラインで上書きすることはできません。たとえば、ユーザー プロパティ

<config>\${jtest.config}</config> として指定されたテスト コンフィギュレーションは、`-Djtest.config` で上書きできます。しかし、<config>builtin://Demo Configuration</config> としてハードコーディングされている場合は上書きできません。

パラメーターの全リストについては、plugins-manual.html の「jtest ゴール」ページを参照してください。

カスタム設定、ライセンス、ルール、テスト コンフィギュレーション、コンパイラ コンフィギュレーションなど、ユーザー レベルでカスタマイズしたすべてのものは、再インストールや更新などの影響を受けないよう、DTP Engine のインストール ディレクトリの外に保存する必要があります。

## 現在の設定の参照

`-showsettings` オプションを使用すると、各設定値がどのファイルで設定されたものかなど、現在の設定およびカスタマイズの内容を出力できます。

### Ant および Maven のパターン

```
<showsettings>true</showsettings>
```

### プロパティ設定のパターン

```
jtest.showsettings=true
```

## 変数の使用

設定値の中で以下の変数を使用できます。

変数	説明	サンプル
analysis_type	有効化されている解析タイプのリストをカンマ区切りで出力します（例：Static, Generation and Execution）	\${analysis_type}
env_var	コロンの後に指定された環境変数の値を出力します。	\${env_var:HOME}
config_name	実行されたテスト コンフィギュレーション名を出力します。	\${config_name}
ntp_project	settings の dtp.project で指定された DTP プロジェクト名を出力します。	\${ntp_project}
project_module	テスト対象プロジェクトのモジュール名を出力します。複数のモジュールが入力として指定された場合、最初にテストされたモジュールの名前が出力され、その後に省略記号 (...) が出力されます。この変数は project.module オプションを使用して設定ファイルで設定できます。	\${module_name}
host_name	ホスト名を出力します。	\${host_name}
user_name	現在のユーザー名を出力します。	\${user_name}
os	オペレーティング システム名を出力します。	\${os}

変数	説明	サンプル
arch	オペレーティング システムのアーキテクチャ名を出力します。	<code>\${arch}</code>
exec_env	実行環境を出力します。この変数は、 <code>\${os}</code> 変数と <code>\${arch}</code> 変数を連結したものです。この変数は <code>exec.env</code> オプションを使用して設定ファイルで設定できます。	<code>\${exec_env}</code>
scontrol_branch	テスト対象プロジェクトのソース管理システムのブランチ名を出力します。複数のブランチが検出された場合、最初のブランチ名が出力され、その後に省略記号 (...) が出力されます。この変数は <code>scontrol.branch</code> オプションを使用して設定ファイルで設定できます。	<code>\${scontrol_branch}</code>
tool_name	ツール名を出力します (Jtest、C++test、dotTEST)	<code>\${tool_name}</code>
jvm_prop	コロン (:) の後に指定された Java vm プロパティの値を出力します。	<code>\${jvm_prop:os.name}</code>

## 設定リファレンス

以下の表は、現在 DTP Engine でサポートされている設定です。

### 基本構成設定

設定	値	説明 / 注意
<code>console.verbosity.level</code>	low normal high	コンソールの冗長レベルを指定します。設定できるオプションは次のとおりです。 low: エラーおよび現在のステップの名前とステータス (done、failed、up-to-date) についての基本情報を表示します。 normal: (デフォルト値) テストと解析の最中にレポートされた問題およびコマンド行も表示します。 high: 警告も表示します。
<code>parallel.mode</code>	disabled auto manual	どのモードをアクティブにするかを決定します。 disabled: 利用できる CPU の中から 1 個だけを使用します。 auto: (デフォルト値) Parasoft Test が並行処理の設定をコントロールします。 manual: ニーズに合わせて並行処理の設定をユーザーが手動で行うことができます。
<code>parallel.no_memory_limit</code>	true false	並行処理に利用されるメモリに、既存のシステム制限を超える制限をかけるかどうかを指定します。 デフォルト値: false

設定	値	説明 / 注意
<code>parallel.free_memory_limit</code>	[percentage]	「アプリケーションで利用できるメモリ全体」に占める「低メモリ状態でキープしておく空きメモリの大きさ」の割合をパーセント値で指定します。このオプションは、他のプロセスが空きメモリを必ず使用できるようにするために使用します。 デフォルト値：25
<code>parallel.max_threads</code>	[number]	同時に実行できる並行スレッドの最大数を指定します。実際の並行スレッドの数は、CPU の数、利用できるメモリ、およびライセンス設定に基づいて決定されます。 デフォルト値は CPU の数と同じです。
<code>file.encoding.mode</code>	default auto user	ファイル エンコーディングの指定方法を定義します。 default：システム プロパティを使用します。 auto： <code>file.encoding.lang</code> で指定された東アジア言語のエンコーディングの自動検出を有効にします。 user： <code>file.encoding.name</code> を使って、明示的にエンコーディング名を指定します。
<code>file.encoding.lang</code>	[code]	<code>file.encoding.mode=auto</code> の場合、言語の数字コードを指定します。有効なコードは次のとおりです。 Japanese = 1 Chinese = 2 Simplified Chinese = 3 Traditional Chinese = 4 Korean = 5
<code>file.encoding.name</code>	[encoding]	<code>file.encoding.mode=user</code> の場合、エンコーディング名を指定します。有効なエンコーディング名は、ASCII-US UTF-8 UTF-16 UTF-16LE UTF-16BE などです。
<code>settings.validation</code>	true false	設定の検証を有効化 / 無効化します。

設定	値	説明 / 注意
settings.rules.file.jtest	path	<p>設定の検証のための追加ルールがあるファイルへのパスを指定します。このファイルは、.properties 形式に従う必要があり、以下の例に従ってルールを記載する必要があります。</p> <pre>engine.path=\$ANY engine.enabled=\$BOOLEAN engine.analysis.deep=\$INTEGER engine.severity.limit=\$REGEXP{[1-5]} engine.verbosity.level=\$REGEXP_IC{(low) (normal) (high)}</pre>

## テスト コンフィギュレーション設定

設定	値	説明 / 注意
configuration.dir.builtin	[path]	ビルトイン テスト コンフィギュレーションのディレクトリへのパスです。
configuration.dir.user	[path]	ユーザー定義テスト コンフィギュレーションのディレクトリへのパスです。
configuration.share.path	[path]	共有テスト コンフィギュレーションの DTP Server share 上のパスです。
jtest.custom.rule.dir	[path to directory]	ユーザー定義のコーディングスタンダード ルールの場所を指定します。デフォルト値は [INSTALL_DIR]/rules/

## Development Testing Platform 設定

設定	値	説明 / 注意
ntp.server	[host]	DTP Server のホスト名を指定します。
ntp.port	[port]	DTP Server のポート番号を指定します。デフォルト値は 443 です。
ntp.user ntp.password	[username] [password]	DTP Server に接続するための認証情報を指定します。
ntp.project	[project_name]	結果をリンクする DTP プロジェクトの名前を指定します。この設定は任意です。
ntp.autoconfig	true false	DTP Server に格納された設定を使って自動設定を有効化します。デフォルト値は false です。
report.ntp.publish	true false	テスト結果を DTP Server にレポートするかどうかを指定します。デフォルト値は false です。
report.ntp.publish.src	off min full	テスト対象のソースコードを DTP Server にパブリッシュするかどうかを指定します。 off: コードは DTP Server にパブリッシュされません。 min: ソースの最小限の部分をパブリッシュします。多くの場合、ソース管理システムへの参照がないソースコード（自動生成されたコードなど）がパブリッシュされます。 full: 指定されたスコープに関連するすべてのソースコードがパブリッシュされます。 report.ntp.publish が有効であれば、デフォルト値は full になり、有効でない場合はデフォルト値は off になります
ntp.share.enabled	true false	Team Server への接続を有効または無効にします。デフォルト値は false です。
jtest.license.use_network	true false	License Service からのライセンスの取得を有効または無効にします。デフォルト値は true です。

設定	値	説明 / 注意
<code>jtest.license.network.type</code>	<code>ntp</code> <code>ls</code>	ネットワーク ライセンスのタイプを設定します。 <code>ntp</code> : ファイル数に基づくライセンスです。ライセンス契約で指定されたファイル数の上限まで解析を行うことができます。 <code>ls</code> : フローティング ライセンス (マシン ロック) です。指定されたマシン数の上限まで利用できます。
<code>jtest.license.local.password</code>	<code>[password]</code>	ローカル ライセンスのパスワードを指定します。
<code>jtest.license.local.expiration</code>	<code>[expiration]</code>	ローカル ライセンスの有効期限を指定します。
<code>jtest.license.network.edition</code>	<code>desktop_edition</code> <code>server_edition</code> <code>custom_edition</code>	License Service から取得するライセンスの種類を指定します。デフォルト値は <code>custom_edition</code> です。
<code>jtest.license.custom_edition_features</code>	<code>[feature_name, ...]</code>	カスタム ライセンス エディションのアクティブにする機能を指定します。
<code>jtest.license.wait.for.tokens.time</code>	<code>[minutes]</code>	使用できるライセンスがない場合にライセンスを待機する時間を指定します。

## スコープおよび作成者の設定

設定	値	説明 / 注意
scope.local	true false	ローカル ユーザーおよびシステムのファイル更新時間に基づいてコード作成者を算定するかどうかを指定します。 デフォルト値: true
scope.scontrol	true false	サポートされるソース管理システムからのデータに基づいてコード作成者を算定するかどうかを指定します。 デフォルト値: false
scope.xmlmap	true false	タスクの割り当てを計算するために、特定のファイルまたはファイルセットのタスクを割り当てる方法を定義した XML マッピング ファイルを使用するかどうかを指定します。 デフォルト値: false
scope.xmlmap.file	[path]	特定のファイルまたはファイルセットのタスクを割り当てる方法を定義した XML マッピング ファイルへのパスを指定します。
authors.ignore.case	true false	作成者で大文字と小文字を区別するかどうかを指定します。 例: true: David と david は同じユーザーと見なされます。 デフォルト値: false
authors.mappings.location	local shared	作成者マッピング ファイルを保存する場所を指定します。デフォルト値は local です。 詳細については authors.user および authors.mapping を参照してください。 shared を指定した場合、DTP share の共有ファイルにマッピングを格納できます。詳細については authors.shared.path を参照してください。

設定	値	説明 / 注意
<code>authors.shared.path</code>	[path]	DTP share 内の作成者マッピング ファイルの場所を指定します。 例： <code>authors.shared.path=xtest/authors_map.txt</code>
<code>authors.user{n}</code>	[user_name, email, full_name]	特定のユーザー名、電子メールアドレス、フルネームで作成者を指定します。 例： <code>authors.user1=dan,dan@parasoft.com,Dan Stowe</code> <code>authors.user2=jim,jim@parasoft.com,Jim White</code>
<code>authors.mapping{n}</code>	[from_user, to_user]	特定の作成者マッピングを指定します。 例： <code>authors.mapping1=old_user,new_user</code> <code>authors.mapping2=broken_user,correct_user</code>

## 抑制の設定

設定	値	説明 / 注意
<code>suppression{n}.file.ext</code>	[ext]	コメントによる抑制をスキャンするファイルの拡張子を指定します。 例： <code>suppression1.file.ext=xml</code> <code>suppression2.file.ext=java</code> <code>suppression.comment</code> 設定でコメントの接頭辞を指定します。
<code>suppression{n}.comment</code>	[comment]	<code>suppression{n}.file.ext</code> 設定で指定されたファイルタイプのコメントの接頭辞を指定します。 例： <code>suppression1.comment=//</code> <code>suppression2.comment=&lt;!--</code> -
<code>suppression{n}.comment.suffix</code>	[comment suffix]	<code>suppression{n}.file.ext</code> 設定で指定されたファイルタイプのコメントの接尾辞を指定します。指定されていない場合、抑制コメントに接尾辞は付加されません。 例： <code>suppression1.comment.suffix=--&gt;</code>
<code>suppression{n}.block.only</code>	true false	<code>suppression{n}.file.ext</code> 設定で指定されたファイルタイプのブロックのみのコメント抑制サポートを有効または無効にします。 デフォルト値：false

## テクニカル サポートの設定

設定	値	説明 / 注意
<code>techsupport.enabled</code>	true false	テクニカル サポート データの自動収集を有効または無効にします。 デフォルト値：false

設定	値	説明 / 注意
logging.verbose	true false	<p>詳細なログ記録を有効または無効にします。</p> <p>詳細ログは local.storage.dir 設定で指定された xtest.log ファイルに保存されます。</p> <p>詳細ログの状態は、セッションを越えて保持され、アプリケーションの起動時に復元されます。ログ ファイルは、一定のサイズ以上に大きくなることはありません。最大サイズに達するたびに、バックアップが作成されます。</p> <p>デフォルト値: false</p>
logging.scontrol.verbose	true false	<p>ソース管理コマンドからの出力を詳細ログに含めるかどうかを指定します。出力にソース コードの一部が含まれることがあります。</p> <p>デフォルト値: false</p>
techsupport.create.on.exit	true false	<p>アプリケーション終了時に、テクニカル サポート アーカイブを自動的に作成するかどうかを指定します。</p> <p>パッケージを自動的に生成するには、techsupport.enabled 設定が有効でなければなりません。</p> <p>デフォルト値: false</p>
techsupport.archive.location	[path]	サポート パッケージを作成するディレクトリを指定します。
techsupport.include.reports	true false	テクニカル サポート パッケージにレポートを含めるかどうかを指定します。

## レポート設定

設定	値	説明 / 注意
session.tag	[name]	テスト セッションの結果をラベル付けするためのタグを指定します。タグは、特定のモジュールに対して実行された特定の解析プロセス固有の識別子です。異なるテストセッションのレポートには、異なるセッションタグを付けるべきです。
build.id	[id]	結果をラベル付けするためのビルド識別子を指定します。ビルドごとに固有の識別子を使用することも、特定のビルド中に実行された複数のテストセッションを同じタグでラベル付けすることもできます。 デフォルトの設定は build-yyyy-MM-dd HH:mm:ss です。
project.module	[name]	プロジェクト モジュールのカスタム名を指定します。個別の実行を説明するために使用できます。指定されていない場合、解析対象として指定されたコードに基づいて、自動的にテスト対象モジュールが検出されます。
exec.env	[env1;env2...]	実行された環境を説明するタグのリストを指定します。タグは、OS (Windows、Linux など)、アーキテクチャ (x86、x86_64)、コンパイラ、ブラウザなどを記述するのに使用できます。exec.env タグを使用すると、セッション全体を説明できます。サービス API 経由で、テストスイート、テスト、またはテストケース レベルで詳細な環境の説明を追加することもできます。
report.location	[path]	レポートが生成される場所を指定します。
report.format	xml html pdf custom CSV	レポートの形式を指定します。複数の形式のレポートを生成するには、カンマで区切った形式のリストを指定します。 デフォルト値：xml
report.custom.extension	[ext]	カスタム レポート形式のための XSL ファイルの拡張子を指定します。 report.format=custom が設定されている場合に report.custom.xsl.file と共に使用します。

設定	値	説明 / 注意
report.custom.xsl.file	[path]	カスタム レポート形式のための XSL ファイルの場所を指定します。 report.format=custom が設定されている場合に report.custom.extension と共に使用します。
report.developer_errors	true false	マネージャー レポートに開発者エラーについての詳細を含めるかどうかを指定します。デフォルト値は true です。
report.developer_reports	true false	管理者に対して HTML サマリー レポートを生成するのに加えて、チーム メンバー全員に対して詳細な HTML レポートを生成するかどうかを指定します。デフォルト値は false です。
report.authors_details	true false	チーム メンバーごとに割り当てられたタスク タイプの数の概要をレポートに含めるかどうかを指定します。デフォルトは true です。
report.contexts_details	true false	テスト中にチェックまたは実行されたファイルの概要をレポートに含めるかどうかを指定します。デフォルトは false です。
report.suppressed_msgs	true false	抑制されたメッセージを HTML レポートに含めるかどうかを指定します。デフォルト設定は false です。
report.metadata	true false	指摘事項に関する追加のメタデータを DTP からダウンロードするかどうかを指定します。既に DTP に存在する指摘事項だけが影響を受けます。また、この設定が効果を持つには、DTP サーバーはメタデータ サービスをサポートしている必要があります。デフォルトは true です。
report.scontrol	off min full	ソース管理システムからの追加情報をレポートに含めるかどうか、また含める情報を指定します。 min: リポジトリ、ファイル パス、およびリビジョン full: min の情報に加えてタスクのリビジョンおよびコメント デフォルトは off です。

設定	値	説明 / 注意
report.associations	true false	テストに関連付けられた要求、欠陥、タスク、および機能要求をレポートに含めるかどうかを指定します。デフォルトは false です。
issue.tracking.tags	[tag1,tag2,...]	課題トラッキング タグのリストを指定します。デフォルトでは、次のタグがサポートされています: pr, fr, task, asset, req
report.assoc.url.[tag]	[url]	HTML レポートに関連成果物へのリンクを生成します。URL は PropertyAttribute 値のプレースホルダー [%ID%] を含むクエリー文字列です。
report.active_rules	true false	テストで有効に設定されているルールをレポートに含めるかどうかを指定します。デフォルトは false です。
report.rules	[url]	静的解析ルールの HTML ファイルを保存するディレクトリを指定します。 例: report.rules=file:///C:/parasoft/gendoc/ report.rules=../gendoc/
report.test_params	true false	レポートにパラメーターの詳細情報を含めるかどうかを指定します。デフォルト設定は false です。
report.coverage.images	[tag1;. . .]	DTP Server でカバレッジ イメージを作成する際に使用されるタグのセットを指定します。 DTP は、レポートごとに最大 3 つのカバレッジ イメージをサポートします。
report.coverage.limit	[limit]	カバレッジの閾値の下限を指定します。この値よりカバレッジが低い場合、レポートで強調表示されます。デフォルトは 40 です。
report.metrics.attributes	[attr1;attr2;...]	メトリクス結果に対する追加属性のリストを指定します。デフォルトでは、次の属性がサポートされています: module, namespace, type, method
report.archive	true false	レポートの zip ファイルへのアーカイブを有効化 / 無効化します。

設定	値	説明 / 注意
report.graph.start_date	[MM/dd/yy]	ある期間にわたって静的解析タスク、テスト実行、カバレッジを記録するトレンドグラフの開始日を指定します。 report.graph.period=[?d ?m ?y] と共に使用します。
report.graph.period	[?d ?m ?y]	ある期間にわたって静的解析タスク、テスト実行、カバレッジを記録するトレンドグラフの終了日を指定します。 report.graph.start_date=[MM/dd/yy] と共に使用します。
report.mail.enabled	true false	開発者と report.mail.cc で指定した送付先に、レポートをメールするかどうかを指定します。 この設定が有効な場合、プロジェクトコードに対して作業している各開発者には、その開発者に関連するエラー/結果だけのレポートが自動的に送信されます。デフォルトは false です。
report.mail.server	[host]	レポートの送信に使用するメールサーバーを指定します。
report.mail.port	[port]	SMTP サーバーのポートを指定します。デフォルトのポートは 25 です。
report.mail.security	[security]	SMTP サーバー接続セキュリティを指定します。STARTTLS および SSL がサポートされています。デフォルト値は STARTTLS です。
report.mail.subject	[subject line]	送信する電子メールの件名を指定します。
report.mail.username report.mail.password report.mail.realm	[user_name] [password] [realm]	SMTP サーバー認証の設定を指定します。 realm の設定は、SASL レalm を使って認証するサーバーにだけ必要です。
report.mail.domain	[domain]	レポートの送信に使用されるメールアドレスのドメインを指定します。
report.mail.time_delay	[time]	迷惑メールの制限を避けるために、電子メールによるレポート送信間の時間の遅延を指定します。
report.mail.from	[email user_name]	送信されるメールの差出人を指定します。

設定	値	説明 / 注意
report.mail.attachments	true false	レポートを添付ファイルとして送信するかどうかを指定します。すべてのコンポーネントは添付ファイルとして含まれます。画像のあるレポートを参照するには、すべての添付ファイルをディスクに保存する必要があります。 デフォルトは false です。
report.mail.compact	trends links	メールで送信する情報を指定します。 trends: トレンド グラフ、概要の表、および他のコンパクト データが電子メールに含まれます。詳細データは含まれません。 links: DTP Server 上にあるレポートへのリンクだけが電子メールに含まれます。 デフォルトでは設定されません。
report.mail.format	html ascii	電子メールのコンテンツ タイプを指定します。デフォルトは html です。
report.mail.cc	[email; ...]	包括的なマネージャー レポートのメール送信先を指定します。複数のアドレスを指定するには、メールアドレスをセミコロン (;) で区切って指定する必要があります。 この設定は通常、マネージャー、アーキテクト、選択された開発者にレポートを送信するのに使用します。
report.mail.include	[email, ...]	開発者レポートを受け取るチームメンバーのアドレスを指定します。複数のアドレスを指定するには、メールアドレスをセミコロン (;) で区切って指定します。 この設定は通常、開発者に自動的にレポートが送信されない場合に、開発者にレポートを送信するために使用します (たとえば、チームがソース管理システムを使用していない場合などです)。 このオプションは、exclude オプションの指定を上書きします。
report.mail.exclude	[email; ...]	レポートの自動送信から除外する電子メールアドレスを指定します。

設定	値	説明 / 注意
report.mail.exclude.developers	true false	report.mail.cc プロパティで明示的に指定されていないチームメンバーに、レポートを送信するかどうかを指定します。個々のチームメンバーにレポートを送信しないようにすることができます。 デフォルトは false です。
report.mail.unknown	[email user_name]	"unknown" に割り当てられたエラー レポートの送信先を指定します。
report.mail.on.error.only	true false	タスクが生成された場合または致命的な例外が発生した場合にだけ管理者にレポートを送信するかどうかを指定します。この設定はチームメンバーへの送信に影響しません。レポートされたタスクに責任があるチームメンバーにだけメールは送信されます。 デフォルトは false です。
report.setup.problems	top bottom hidden	セットアップの問題に関するセクションをレポートのどこに表示するかを指定します。デフォルトは bottom です。
report.setup.problems.category_limit	[numerical value]	1 つのセットアップの問題カテゴリに表示されるメッセージ数の上限を指定します。 デフォルト値: 10
report.setup.problems.display_limit	[numerical value]	HTML レポートのセットアップの問題カテゴリに表示される合計メッセージ数の上限を指定します。 デフォルト値: 100
report.setup.problems.console	true false	セットアップの問題をコンソールに出力するかどうかを指定します。 デフォルトは true です。
report.ue_coverage_details_htmls	LC DC	行ごとのカバレッジ詳細情報が付加されたソースコードのレポートに、テストの HTML レポートからリンクする場合、含めるカバレッジの種類を指定します。 LC: 行カバレッジ DC: 判断文 (Decision) カバレッジ
report.separate_vm.xml	[size]	レポートの生成に使用されるメモリの量を指定します。デフォルト値: 1024M

設定	値	説明 / 注意
report.separate_vm	true false	仮想マシンごとに別のレポートを生成するかどうかを指定します。 デフォルト値: false
report.separate_vm.launch.file	[path]	レポート生成時に使用する起動ファイルへのパスを指定します。
dupcode.sorting.mode	oldest newest paths	重複コードの指摘での要素のソート順を指定します。 oldest: 一番古い結果が一番上に表示されます。 newest: 一番新しい結果が一番上に表示されます。 paths: フルパス名のアルファベット昇順 (A から Z) で表示されます。 デフォルト値: paths
report.coverage.version	1 2	XML カバレッジ レポートのバージョンを指定します。 1: 標準バージョンが使用されます。 2: XML レポートのサイズが最適化されます。 デフォルト値は 1 です。

## 汎用ソース コントロール システムの設定

設定名	値	説明 / 注意
scontrol.timeout	[seconds]	ソース管理システム操作時のタイムアウトを指定します。デフォルト値: 60
scontrol.branch	[name]	テスト対象ブランチのカスタム名を指定します。個別の実行を説明するために使用できます。指定されていない場合、解析対象として指定されたコードに基づいて、自動的にテスト対象ブランチが検出されます。

## AccuRev ソース管理システムの設定

設定名	値	説明 / 注意
scontrol.rep{n}.type	accurev	AccuRev リポジトリ タイプの識別子。
scontrol.accurev.exec	[path]	外部クライアント実行モジュール (accurev). へのパス。

設定名	値	説明 / 注意
<code>scontrol.rep{n}.accurev.host</code>	[host]	AccuRev サーバー ホスト。
<code>scontrol.rep{n}.accurev.port</code>	[port]	AccuRev サーバー ポート。 デフォルト ポートは 1666。
<code>scontrol.rep{n}.accurev.login</code>	[login]	AccuRev のユーザー名。
<code>scontrol.rep{n}.accurev.password</code>	[password]	AccuRev のパスワード。

## ClearCase ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	ccase	ClearCase リポジトリ タイプ の識別子。
<code>scontrol.ccase.exec</code>	[path]	外部クライアント実行モジュール (cleartool) へのパス。
<code>scontrol.rep{n}.ccase.vob</code>	[path]	VOB のマウントポイント。ユー ザーが VOB にアクセスするパ スです。 例： scon- trol.rep.ccase.vob=X:\myvob scontrol.rep.ccase.vob=/vobs/ myvob
<code>scontrol.rep{n}.ccase.vob_tag</code>	[tag]	ClearCase ネットワーク リー ジョン内での VOB の一意なタ グ

## CVS ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	<code>cvs</code>	CVS リポジトリ タイプの識別子。
<code>scontrol.rep{n}.cvs.root</code>	<code>[root]</code>	完全な CVSROOT の値。
<code>scontrol.rep{n}.cvs.pass</code>	<code>[password]</code>	プレーン テキスト パスワード または暗号化されたパスワード。 暗号化されたパスワードは、 .cvspass ファイル中のパス ワードと同じでなければなりません。 CVS の場合、ユーザーのホーム ディレクトリの .cvspass の値 を使用します。 CVSNT の場合、レジストリの HKEY_CURRENT_USER\Soft ware\Cvsnt\cvspass の値を使 用します。 "cvs login" を使ってコマン ドラインから初めて CVS リポ ジトリにログインした場合、パ スワードはレジストリに保存さ れます。このパスワードを取得 するには、regedit を使ってレ ジストリを参照し、 HKEY_CURRENT_USER- >CVSNT> cvspass キーの下 の値を探します。この値には完 全なログイン名 (:pserver:exampleA@examp leB:/exampleC) と暗号化され たパスワード値が表示されます。
<code>scontrol.rep{n}.cvs.useCustomSSHCredentials</code>	<code>true</code> <code>false</code>	EXT/SSH 接続のために CVS の ログイン名とパスワードを使用 するかどうかを指定します。デ フォルトは <code>false</code> です。
<code>scontrol.rep{n}.cvs.ext.server</code>	<code>[cvs]</code>	EXT モードで CVS サーバーに 接続している場合、このオブ ションは、サーバー サイドでど の CVS アプリケーションを起 動するかを指定します。 CVS_SERVER 変数と同じ意味 を持ちます。cvs がデフォルト です。
<code>scontrol.rep{n}.cvs.ssh.loginname</code>	<code>[login]</code>	外部プログラムを使ってログイン を提供できる場合に、SSH 接 続のためのログイン名を指定し ます。

設定名	値	説明 / 注意
<code>scontrol.rep{n}.cvs.ssh.password</code>	[password]	SSH 接続のためのパスワードを指定します。
<code>scontrol.rep{n}.cvs.ssh.keyfile</code>	[file]	キー認証を使って SSH 接続を確立するための秘密鍵ファイルを指定します。
<code>scontrol.rep{n}.cvs.ssh.passphrase</code>	[passphrase]	キー認証メカニズムを使った SSH 接続のためのパスフレーズを指定します。
<code>scontrol.rep{n}.cvs.useShell</code>	true false	CVS リポジトリへの接続を確立するために、外部プログラム (CVS_RSH) を有効にします。デフォルトは false です。
<code>scontrol.rep{n}.cvs.ext.shell</code>	[path]	CVS_RSH プログラムとして使用される実行ファイルへのパスを指定します。コマンドラインパラメーターは <code>cvs.ext.params</code> プロパティで指定すべきです。
<code>scontrol.rep{n}.cvs.ext.params</code>	[parameters]	外部プログラムに渡すパラメーターを指定します。次のマクロを使って、値をコマンドラインパラメーターに展開できます。これらのマクロは大文字と小文字を区別します。 {host} リポジトリ ホスト {port} ポート {user} cvs ユーザー {password} cvs パスワード {extuser} パラメーター cvs.ssh.loginname {extpassword} パラメーター cvs.ssh.password {keyfile} パラメーター cvs.ssh.keyfile {passphrase} パラメーター cvs.ssh.passphrase

## Git ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	git	Git リポジトリ タイプの識別子。
<code>scontrol.git.exec</code>	[path]	Git の実行モジュールへのパス。 設定しない場合、コマンドはパス上にあると見なされます。
<code>scontrol.rep{n}.git.url</code>	[url]	リモート リポジトリ (例 <code>git://hostname/repo.git</code> )
<code>scontrol.rep{n}.git.workspace</code>	[path]	ローカルの git リポジトリがあるディレクトリ。

## Mercurial ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	hg	Mercurial リポジトリ タイプの識別子。
<code>scontrol.hg.exec</code>	[path]	外部クライアント実行モジュールへのパス。デフォルト値: hg
<code>scontrol.rep{n}.hg.url</code>	[url]	リモート リポジトリの URL (例: <code>http://hostname/path</code> )
<code>scontrol.rep{n}.hg.workspace</code>	[path]	ローカル Mercurial リポジトリがあるディレクトリ

## Perforce ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	perforce	Perforce リポジトリ タイプの識別子。
<code>scontrol.perforce.exec</code>	[path]	外部クライアント実行モジュール (p4) へのパス。
<code>scontrol.rep{n}.perforce.host</code>	[host]	Perforce のサーバー ホスト。
<code>scontrol.rep{n}.perforce.port</code>	[port]	Perforce のサーバー ポート。デフォルトは 1666。
<code>scontrol.rep{n}.perforce.login</code>	[login]	Perforce ユーザー名。
<code>scontrol.rep{n}.perforce.password</code>	[password]	パスワード。チケットを使用して認証する場合、指定は任意です。
<code>scontrol.rep{n}.perforce.client</code>	[client]	P4CLIENT 環境変数またはそれに相当するもので指定されているクライアント ワークスペース名。指定された、ワークスペースのルート ディレクトリはローカル マシンに対して適切に構成する必要があります。

## Serena Dimensions ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	serena	Serena Dimensions リポジトリのタイプ識別子。
<code>scontrol.serena.dmroot</code>	[path]	Serena Dimensions の実行モジュールへのパス。 例： C:\Program Files (x86)\Serena\Dimensions 2009 R2\CM\
<code>scontrol.rep{n}.serena.login</code>	[login]	ユーザー名。
<code>scontrol.rep{n}.serena.password</code>	[password]	パスワード。
<code>scontrol.rep{n}.serena.host</code>	[host]	Serena Dimensions のサーバー ホスト名。
<code>scontrol.rep{n}.serena.dbname</code>	[name]	使用するデータベースの名前。
<code>scontrol.rep{n}.serena.dbconn</code>	[connection]	データベースへの接続文字列。
<code>scontrol.rep{n}.serena.locale</code>	[locale]	使用する言語。 例 : en_US

設定名	値	説明 / 注意
<code>scontrol.rep{n}.serena.mapping</code>	[mapping]	デフォルトの作業エリア以外の場所にプロジェクトをダウンロードまたは移動した場合、このオプションを使って、Serena リポジトリのプロジェクト（またはストリーム）とローカルプロジェクトとのマッピングを指定します。 デフォルト作業エリアで作業している場合、マッピングを定義する必要はありません。

## StarTeam ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	starteam	StarTeam リポジトリ タイプの識別子。
<code>scontrol.rep{n}.starteam.host</code>	[host]	StarTeam のサーバー ホスト。
<code>sscontrol.rep{n}.starteam.port</code>	[port]	StarTeam のサーバー ポート。 デフォルトは 49201。
<code>scontrol.rep{n}.starteam.login</code>	[login]	ログイン名。
<code>scontrol.rep{n}.starteam.password</code>	[password]	暗号化していないパスワード。

設定名	値	説明 / 注意
scontrol.rep{n}.starteam.path	[path]	<p>大規模なマルチ プロジェクト リポジトリで作業している場合、作業対象のプロジェクト、ビュー、またはフォルダーを指定してパフォーマンスを向上させることができます。</p> <p>次のものを指定できます。</p> <p>単純なプロジェクト名 (リポジトリパスを検索するときにすべてのビューがスキャンされます)。</p> <p>プロジェクト/ビュー (指定のビューだけがスキャンされます)。</p> <p>プロジェクト/ビュー/フォルダー (指定の StarTeam フォルダーがスキャンされます)。</p> <p>例 :</p> <pre>scontrol.rep.starteam.path=proj1 scontrol.rep.starteam.path=proj1/view1 scontrol.rep.starteam.path=proj1/view1/folderA scontrol.rep.starteam.path=proj1/view1/folderA/folderB</pre>
scontrol.rep{n}.starteam.workdir	[path]	<p>scontrol.rep.starteam.path の設定で StarTeam のビューまたはフォルダーを指定する場合、このプロパティを使って、次のものを指定できます。</p> <p>選択されたビューのルート フォルダーに対する新しい作業ディレクトリ (パスがビューを表す場合)</p> <p>選択されたフォルダーに対する新しい作業ディレクトリ (パスがフォルダーを表す場合)</p> <p>例 :</p> <pre>scontrol.rep.starteam.workdir=c:\\storage\\dv scontrol.rep.starteam.workdir=/home/storage/dv</pre>

## Subversion ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	<code>svn</code>	Subversion リポジトリ タイプの識別子。
<code>scontrol.svn.exec</code>	<code>[path]</code>	外部クライアント実行モジュール (svn) へのパス。
<code>scontrol.rep{n}.svn.url</code>	<code>[url]</code>	Subversion URL はプロトコル、サーバー名、ポートおよび開始リポジトリ パスを指定します。 例： <code>svn:// buildmachine.foobar.com/ home/svn</code>
<code>scontrol.rep{n}.svn.login</code>	<code>[login]</code>	ログイン名。
<code>scontrol.rep{n}.svn.password</code>	<code>[password]</code>	暗号化していないパスワード。

## Synergy/CM ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	<code>synergy</code>	Synergy/CM リポジトリ タイプの識別子。
<code>scontrol.synergy.exec</code>	<code>[path]</code>	外部クライアント実行モジュール (ccm) へのパス
<code>scontrol.rep{n}.synergy.host</code>	<code>[host]</code>	synergy/cm エンジンが稼動するコンピューター。このプロパティを指定しない場合、ローカル ホストが使用されます。Web モードの場合、ホストはプロトコルとポートを含む有効な Synergy Web URL でなければなりません (例 <code>http:// synergy.server:8400</code> )。
<code>scontrol.rep{n}.synergy.dbpath</code>	<code>[path]</code>	synergy データベースの絶対パス。 例： <code>\\host\db\name</code>
<code>scontrol.rep{n}.synergy.projspec</code>	<code>[specification]</code>	プロジェクト名とバージョンを格納した Synergy プロジェクト仕様。 例： <code>name-version</code>
<code>scontrol.rep{n}.synergy.login</code>	<code>[login]</code>	Synergy ユーザー名。

設定名	値	説明 / 注意
<code>scontrol.rep{n}.synergy.password</code>	[password]	Synergy の暗号化していないパスワード。
<code>scontrol.rep{n}.synergy.port</code>	[port]	Synergy のポート。
<code>scontrol.rep{n}.synergy.remote_client</code>	[client]	(UNIX のみ) ccm をリモートクライアントとして開始するかどうかを指定します。デフォルトは false です。任意。このプロパティは Web モードでは使用されません。
<code>scontrol.rep{n}.synergy.local_dbpath</code>	[path]	リモートクライアントセッションを実行するときにデータベース情報をコピーするコピー先のパスを指定します。このプロパティが null の場合、デフォルトの場所が使用されます。このプロパティは Web モードでは使用されません。

## Microsoft Team Foundation Server ソース管理システムの設定

設定名	値	説明 / 注意
<code>scontrol.rep{n}.type</code>	tfs	TFS リポジトリ タイプの識別子。
<code>scontrol.rep{n}.tfs.url</code>	[url]	TFS リポジトリの URL (例: <code>http://localhost:8080/tfs</code> )
<code>scontrol.rep{n}.tfs.login</code>	[login]	TFS ユーザー名。
<code>scontrol.rep{n}.tfs.password</code>	[password]	TFS パスワード。

## Microsoft Visual SourceSafe ソース管理システムの設定

設定名	値	説明 / 注意
scontrol.rep{n}.type	vss	Visual SourceSafe リポジトリタイプの識別子。
scontrol.vss.exec	[path]	外部クライアント実行モジュール (ss) へのパス。
scontrol.rep{n}.vss.ssdire	[path]	リポジトリ データベースのパス。
scontrol.rep{n}.vss.projpath	[path]	VSS プロジェクトパス。
scontrol.rep{n}.vss.login	[login]	VSS ログイン。
scontrol.rep{n}.vss.password	[password]	VSS パスワード。

### その他の設定

設定名	値	説明 / 注意
jtest.fail	true false	違反がレポートされた場合、ゼロではない終了コードを返してビルドを失敗させます。
jtest.show.settings	true false	現在の設定とカスタマイズを出力します。各構成エントリ (たとえば .properties ファイル) のソースを示します。
jtest.data.additional	[path]	追加の json.data ファイルのパス。ファイルが複数の場合、パスのリストをカンマで区切って指定します。

# 統合

- ソース管理システムとの統合
- ビルド システムとの統合
- 外部の解析ツールとの統合
- IDE での DTP Engine の使用
- CI ツールとの統合

# ビルドシステムとの統合

Jtestcli.exe は、ビルドシステム プラグインによって作成される `jtest.data.json` ファイルを基に実行されます。`jtest.data.json` ファイルの構造については 8 ページの「テスト データの場所の指定」を参照してください。Maven、Ant、または Gradle の問題のトラブルシューティングなど、プラグインの詳細については [manuals/ja/plugins-manual/iindex.html](https://manuals/ja/plugins-manual/iindex.html) を参照してください。

## Jtest 9.5 ユーザー向けの追加情報

Static Analysis Engine for Java のリリースによって、Jtest 9.5 と比較してビルド自動化システムのサポートが大幅に強化されました。Static Analysis Engine は、Jtest 9.x とは異なる方法でビルドシステムと統合されます。Static Analysis Engine は Jtest と同じ機能を持っていますが、現行バージョンの Static Analysis Engine では、解析を実行するために Eclipse を使用したり、プロジェクトをワークスペースにインポートする必要がなくなりました。この変更により、実行時間が大幅に短縮されます。

これに伴い、コマンドライン ツールも変更されました。結果として、名前の一貫性を保つためにビルドシステム プラグインのオプションが変更されました。

# 外部の解析ツールとの統合

## Checkstyle

DTP Engine は、Checkstyle 5.5、5.6、5.7、および 6.5 での静的解析をサポートしています。

Checkstyle ルールを有効にし、このフォルダーにドロップしたプラグインを使用するには、`$JTEST_INSTALL_DIR/etc/framework.properties` 構成ファイルの設定を変更する必要があります。

`felix.auto.start.4` セットの最初のエン트리として `[INSTAL_DIR]/plugins/builtin/framework/` にある次の jar ファイルを追加します。

```
org.apache.felix.fileinstall-3.4.0.jar
```

## Checkstyle のダウンロードとインストール

1. 次のウェブ サイトから Checkstyle をダウンロードします。  
<http://sourceforge.net/projects/checkstyle/files>
2. Checkstyle を展開し、インストールします。
3. `$HOME/integration/checkstyle` ディレクトリにある `configs`、`plugins`、`rules` フォルダをインストール ディレクトリにコピーします。
4. `jtestcli.properties` ファイルを編集して次のプロパティを追加します。

```
jtest.analyzer.checkstyle.enabled=true
jtest.analyzer.checkstyle.dir=CHECKSTYLE/INSTALLATION/DIRECTORY
rules.provider_cs.analyzer=com.puppycrawl.tools.checkstyle
rules.provider_cs.data=${jtest.home}/rules/csrules.xml
```

## Checkstyle を使用したコードの解析

Checkstyle ユーザー定義テスト コンフィギュレーションを指定して Checkstyle での解析を実行します。

<code>user://Checkstyle</code>	すべてのルールが有効
<code>user://Checkstyle Recommended</code>	Checkstyles 推奨ルールが有効

## オプションのカスタマイズ

以下のプロパティを構成して Checkstyle との統合をカスタマイズできます。

```
# Custom limit for number of files to be tested in a single executed Checkstyle process. Default is 50.
# However, if subsequent files are placed in the same directory, they will be still scheduled to that
# process unless the "hard" limit is reached (twice the "regular" limit).
jtest.analyzer.checkstyle.files.limit=50

# Custom timeout (seconds) for the Checkstyle process to start. Default is 5 seconds.
jtest.analyzer.checkstyle.launch.timeout=5

# Custom timeout (seconds) for the Checkstyle process to complete. Default is 60 seconds.
jtest.analyzer.checkstyle.timeout=60

# Custom launch file for Checkstyle process. Default is taken from the Checkstyle analyzer jar file.
# If one cannot be found, attempts to use file 'checkstyle.ini' from current working directory.
jtest.analyzer.checkstyle.launch.file=CUSTOM/LAUNCH/FILE
```

\$HOME/integration/checkstyle ディレクトリにある rulemap.txt ファイルには、テスト コンフィギュレーションで使用されるルール ID が、どのように Checkstyle のルール タイプにマッピングされるかに関する情報が記載されています。

## FindBugs

Static Analysis Engine は、FindBugs 2.0.2 および 2.0.3 での静的解析をサポートしています。

### FindBugs のダウンロードとインストール

1. 次のウェブ サイトから FindBugs をダウンロードします。  
<http://findbugs.sourceforge.net/downloads.html>
2. FindBugs を展開し、インストールします。
3. \$HOME/integration/findbugs ディレクトリにある configs、plugins、rules フォルダをインストール ディレクトリにコピーします。
4. jtestcli.properties ファイルを編集して次のプロパティを追加します。

```
jtest.analyzer.findbugs.enabled=true
jtest.analyzer.findbugs.dir=FINDBUGS/INSTALLATION/DIRECTORY
rules.provider_fb.analyzer=edu.umd.cs.findbugs
rules.provider_fb.data=${jtest.home}/rules/fbrules.xml
```

### FindBugs を使用したコードの解析

FindBugs ユーザー定義テスト コンフィギュレーションを指定して FindBugs での解析を実行します。

user://FindBugs	すべてのルールが有効
user://FindBugs Recommended	FindBugs 推奨ルールが有効

FindBugs はコンパイル済みクラスを必要とするため、解析の前にプロジェクトをビルドする必要があります。

## オプションのカスタマイズ

以下のプロパティを構成して FindBugs との統合をカスタマイズできます。

```
# Custom limit for number of files to be tested in a single executed FindBugs process.
Default is reasonably high.
jtest.analyzer.findbugs.files.limit=1000

# Custom timeout (seconds) for the FindBugs process to start. Default is 5 seconds. /
jtest.analyzer.findbugs.launch.timeout=5

# Custom timeout (seconds) for the FindBugs process to complete. Default is 360 seconds. /
jtest.analyzer.findbugs.timeout=360

# Custom launch file for FindBugs process. Default is taken from the FindBugs analyzer jar
file.
# If one cannot be found, attempts to use file 'findbugs.ini' from current working directory.
jtest.analyzer.findbugs.launch.file=CUSTOM/LAUNCH/FILE
```

\$HOME/integration/findbugs ディレクトリにある rulemap.txt ファイルには、テスト コンフィギュレーションで使用されるルール ID が、どのように FindBugs のルール タイプにマッピングされるかに関する情報が記載されています。

# ソース管理システムとの統合

DTP Engine はソース管理システムから情報を収集し、そのデータを使用して違反の担当者を割り当てたり、時刻や変更履歴に基づいて解析対象ファイルをフィルターしたり、管理対象のファイルの情報を DTP Server にレポートしたりできます。Jtest 9.5 以降のインターフェイスを使用してソース管理システムとの統合を設定するには、次の操作を行います。

1. IDE で [Parasoft] > [設定] をクリックし、[ソース管理] をクリックします。
2. リポジトリおよびソース管理クライアントの設定を行い、[適用] をクリックします。
3. 設定パネルのメニューで [スコープと作成者] をクリックします。
4. [ソース管理システム (更新作成者)] を使ってスコープを計算] オプションをオンにし、[適用] をクリックします。
5. 設定パネルのメニューで [Parasoft] をクリックします。
6. [共有] をクリックしてローカル設定ファイルへのエクスポート パネルを開きます。
7. [ソース管理]、[スコープと作成者] およびその他の保存対象のオプションを選択します。
8. 場所を指定して [OK] をクリックします。
9. 設定ファイルに次の行を追加します。この行により、ソース管理システムの詳細な情報が保存され、レポートされます。

```
report.scontrol=min
```

10. このファイルをコマンドラインで指定するか、DTP Server プロジェクトの管理パネル (Parasoft Test 設定タブ) に設定をコピーします。
11. 解析を実行します。

# CI ツールとの統合

## Jenkins との統合

DTP Engine for Java は、継続的インテグレーション ツールの Jenkins と統合することができます。Parasoft Findings Plugin for Jenkins を使用すると、静的解析とテスト結果を、トレンドグラフと警告として可視化できます。

Parasoft Findings Plugin は直接 Jenkins で利用できます。詳細については [Parasoft Findings Plugin](#) を参照してください。

Parasoft Findings Plugin のソース ファイルは GitHub からダウンロードできます。詳細については [Parasoft Findings Plugin Project](#) を参照してください。Parasoft Findings Plugin のリビルド方法についてさらに情報が必要な場合、Parasoft サポートまでお問い合わせください。

# ヘルプの参照

コマンドラインで使用方法に関する情報を表示するには、-help スイッチを使用します。

```
jtestcli.exe -help
```

## テクニカル サポート

テクニカル サポート用パッケージを作成するよう DTP Engine を設定できます。.properties コンフィギュレーション ファイルに次の設定を追加します。

```
techsupport.enabled=true  
techsupport.create.on.exit=true  
techsupport.archive.location=[OUTPUT DIRECTORY]
```

解析終了時、テクニカル サポート パッケージが出力ディレクトリに作成されます。

パッケージの場所を指定するには、バックスラッシュをエスケープする必要があります。

```
C:\\Project\\MailSystem\\Report
```

# サードパーティのコンテンツ

DTP Engine for Java には、サードパーティのコンテンツが含まれています。以下の表は、各項目の名称とライセンス契約についてまとめたものです。詳細についてはライセンス名をクリックしてください。

項目名	ライセンス
commons-collections.jar	Apache License 2.0
commons-vfs.jar	Apache License 2.0
avalon-framework.jar	Apache License 2.0
batik-all.jar	Apache License 2.0
fop.jar	Apache License 2.0
chardet.jar	Mozilla Public License
bcprov.jar	MIT License
saxon.jar	Mozilla Public License
jfreechart.jar	GNU LGPL License
jcommon.jar	GNU LGPL License
cvslib.jar	CDDL License
javax.xml.stream_1.0.1.jar	Eclipse Public License
javax.activation_1.1.1.jar	Apache License 2.0
jakarta-log4j.jar	Apache License 2.0
xmlgraphics-commons.jar	Apache License 2.0
fst.jar	Apache License 2.0
truezip.jar	Apache License 2.0
jjawin.jar	DevelopMentor OpenSource Software License
trilead-ssh2.jar	Trilead AG License
javanet.staxutils_1.0.0.jar	BSD License
commons-codec.jar	Apache License 2.0
commons-httpclient.jar	Apache License 2.0
org.apache.commons.io_1.4.0.v20081110-1000.jar	Apache License 2.0
org.apache.commons.logging_1.1.3.jar	Apache License 2.0

項目名	ライセンス
httpclient.jar	Apache License 2.0
httpcore.jar	Apache License 2.0
httpclient.jar	Apache License 2.0
httpmime.jar	Apache License 2.0
org.apache.jcs_1.3.4.jar	Apache License 2.0
org.codehaus.stax2_3.2.4.jar	Apache License 2.0
org.json_1.0.0.v201507290100.jar	JSON License
javax.mail_1.5.0.jar	CDDL License
org.suigeneris.jrcs.diff_0.4.2.jar	GNU LGPL License
org.apache.felix.scr-1.6.2.jar	Apache License 2.0
osgi.core-5.0.0.jar	Apache License 2.0
antlr-2.7.7.jar	ANTLR 4 License (BSD)
asm	ASMDEX License
asm-analysis	ASMDEX License
asm-commons	ASMDEX License
asm-tree	ASMDEX License
asm-util	ASMDEX License
c3p0-0.9.5.jar	GNU LGPL License 2.1, Eclipse Public License 1.0
com.esotericsoftware.minlog_1.2.0.jar	BSD License
commons-collections-3.2.2.jar	Apache License 2.0
commons-compress-1.11.jar	Apache License 2.0
commons-configuration_1.9.0.jar	Apache License 2.0
commons-dbutils-1.6.jar	Apache License 2.0
commons-exec-1.1.jar	Apache License 2.0
commons-io-2.2.jar	Apache License 2.0
commons-io_2.4.0.jar	Apache License 2.0
commons-lang-2.6.jar	Apache License 2.0
commons-lang3-3.1.jar	Apache License 2.0
com.sun.xml.bind.2.2.7.jar	CDDL and LGPL 2.1
dom4j-1.6.1.jar	BSD License

項目名	ライセンス
doxia-core-1.2.jar	Apache License 2.0
doxia-decoration-model-1.2.jar	Apache License 2.0
doxia-logging-api-1.2.jar	Apache License 2.0
doxia-module-xhtml-1.2.jar	Apache License 2.0
doxia-sink-api-1.2.jar	Apache License 2.0
doxia-site-renderer-1.2.jar	Apache License 2.0
gson-2.2.4.jar	Apache License 2.0
guava_18.0.0.jar	Apache License 2.0
h2-1.4.186.jar	Mozilla Public License 2.0, or Eclipse Public License 1.0
hibernate-c3p0-4.3.5.Final.jar	GNU LGPL License 2.1
hibernate-commons-annotations-4.0.4.Final.jar	GNU LGPL License 2.1
hibernate-core-4.3.5.Final.jar	GNU LGPL License 2.1
hibernate-jpa-2.1-api-1.0.0.Final.jar	Eclipse Public License 1.0, Eclipse Distribution License 1.0
jandex-1.1.0.Final.jar	Apache License 2.0
javassist-3.18.1-GA.jar	Mozilla Public License 1.1, LGPL 2.1, Apache License 2.0
javassist-3.18.2-GA.jar	Mozilla Public License 1.1, LGPL 2.1, Apache License 2.0
jaxb-api_2.2.7.jar	CDDL 1.1,GPL 2
jboss-logging-3.1.3.GA.jar	Apache License 2.0
jboss-transaction-api_1.2_spec-1.0.0.Final.jar	CDDL, GPL 2.0 with the Class- path Exception
jsr305-2.0.3.jar	Apache License 2.0
junit-4.11.jar	Common Public License 1.0
jython_2.5.3.jar	Jython Software License
kryo_2.24.0.jar	BSD License
lucene-core.jar	Apache License 2.0
maven-artifact-2.1.0.jar	Apache License 2.0
maven-reporting-api-3.0.jar	Apache License 2.0

項目名	ライセンス
maven-reporting-impl-2.2.jar	Apache License 2.0
mchange-commons-java-0.2.9.jar	GNU LGPL License 2.1, Eclipse Public License 1.0
mockito-all-1.9.5.jar	MIT License
objenesis-2.1.jar	Apache License 2.0
org.abego.treelayout.core_1.0.1.jar	BSD 3-Clause 'New' or 'Revised' License (BSD-3-Clause)
org.antlr.4-annotations_4.2.1.jar	ANTLR 4 License (BSD)
org.antlr.4-runtime_4.2.1.jar	ANTLR 4 License (BSD)
org.apache.felix.fileinstall-3.5.0.jar	Apache License 2.0
org.apache.felix.gogo.command-0.14.0.jar	Apache License 2.0
org.apache.felix.gogo.runtime-0.12.1.jar	Apache License 2.0
org.apache.felix.gogo.shell-0.10.0.jar	Apache License 2.0
org.apache.felix.main-4.4.1.jar	Apache License 2.0
org.apache.felix.scr-1.8.2.jar	Apache License 2.0
org.eclipse.core.content-type_3.5.100.jar	Eclipse Public License 1.0
org.eclipse.core.filesystem_1.6.0.jar	Eclipse Public License 1.0
org.eclipse.core.jobs_3.8.0.jar	Eclipse Public License 1.0
org.eclipse.core.resources_3.11.0.jar	Eclipse Public License 1.0
org.eclipse.core.runtime_3.12.0.jar	Eclipse Public License 1.0
org.eclipse.equinox.common_3.8.0.jar	Eclipse Public License 1.0
org.eclipse.equinox.preferences_3.6.0.jar	Eclipse Public License 1.0
org.eclipse.jdt.core_3.12.0.jar	Apache License 2.0
org.eclipse.osgi_3.11.0.jar	Eclipse Public License 1.0
org.eclipse.text_3.6.0.jar	Eclipse Public License 1.0
org.objenesis_2.1.0.jar	Apache License 2.0
org.restlet	Apache License 2.0
org.restlet.ext.json	Apache License 2.0
org.restlet.ext.simple	Apache License 2.0

項目名	ライセンス
org.restlet.lib.org.json	Apache License 2.0
org.simpleframework	Apache License 2.0
oro_2.0.8.jar	Apache License 1.1
plexus-archiver-2.10.1.jar	Apache License 2.0
plexus-il8n-1.0-beta-10.jar	Apache License 2.0
plexus-io-2.6.jar	Apache License 2.0
plexus-utils-2.1.jar	Apache License 2.0
plexus-velocity-1.1.8.jar	Apache License 2.0
powermock-mockito-1.5.5-full.jar	Apache License 2.0
servlet-api.jar	Apache License 2.0
spell.jar	Apache License 2.0
stax2-api-3.1.1.jar	BSD License
velocity-1.7.jar	Apache License 2.0
woodstox-core-lgpl-4.2.0.jar	GNU LGPL License 2.1
xmllpull_1.1.3.1.jar	Public Domain
xpp3.min_1.1.0.4c.jar	Public Domain
xstream_1.4.4.jar	BSD style