



**Concerto および
Development Testing PlatformParasoft
と 9.x レガシー 機能**

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 IT IS 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.

1.0 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.

2.0 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 or annual (as specified in an Order Instrument), 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 or Your affiliates' 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. Notwithstanding the foregoing, You may allow Your affiliates and third party contractors to use the Software solely for Your benefit; provided, however, that You shall remain fully responsible for such affiliates' and contractors' compliance with this Agreement.

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.

3.0 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.

4.0 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.

5.0 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.

6.0 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.

7.0 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.

8.0 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.

9.0 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 or Parasoft's authorized distributor 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 or Parasoft's authorized distributor to one designated support coordinator of Yours (and one backup coordinator) per Your location, by telephone, email, or 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 (indicated by the first number to the right of the first decimal point) 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.

10.0 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 10.3.

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.

Parasoft Corporation
101 East Huntington Drive, 2nd Floor
Monrovia, CA 91016 USA
+1 (626) 256-3680
+1 (888) 305-0041 (USA only)
+1 (626) 256-9048 (Fax)
info@parasoft.com
<http://www.parasoft.com>

All brand and product names are trademarks or registered trademarks of their respective holders.

Copyright 1993-2016
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。

1.0 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。

2.0 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こと、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、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つたとしても、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。

3.0 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。

4.0 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。

5.0 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に不具合を书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。

6.0 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、賠償するものとします。(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。

7.0 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係わらず、適用されます。

8.0 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。

9.0 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年間が適用されます。

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の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。

10.0 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弁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「商用コンピューターソフトウェア」および「商用コンピューター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ジェ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。

Parasoft Corporation

101 East Huntington Drive, 2nd Floor

Monrovia, CA 91016 USA

+1 (626) 256-3680

+1 (888) 305-0041 (USA only)

+1 (626) 256-9048 (Fax)

info@parasoft.com

<http://www.parasoft.com>

すべての商標ならびに製品名は、それぞれの商標権者の商標マークまたは登録商標マークです。

Copyright 1993-2016

Parasoft Corporation

101 East Huntington Drive, 2nd Floor

Monrovia, CA 91016 USA

目次

はじめに

Eclipse および Visual Studio との統合	4
--------------------------------------	---

ウィジェット

ビルド結果ウィジェット	7
コード ウィジェット	9
欠陥ウィジェット	11
Project Center ウィジェット	13
静的解析 (9.x) ウィジェット	15
テスト (9.x) ウィジェット	19

Parasoft Test ツールのレポート (9.x 以前)

レポートの参照	30
スケジュール、コスト、品質の管理	31
静的解析レポート	38
[単体テスト] レポート	51
[ソース コードのチェックイン] レポート	59
ビルド レポート	62
コード レビュー レポート	68
[ツールの使用] レポート	73
[テストの概要] レポート	82
[日付別テスト] レポート	84
[カテゴリ別エラー] レポート	85
[重要度別エラー] レポート	94
[最近のテスト ログ] レポート	98
変更ベースのテスト レポート	105
コードメトリクス レポート	108
セキュリティ レポート	116
ポリシー レポート	121
プロジェクト ポートフォリオ	122
コードベースのサイズ	125
ビルド結果	132
テスト (ファイル)	133
欠陥および改善レポート	144
手動テスト セッション	149
カバレッジ レポート	151

Policy Center (レガシー) ユーザーズ ガイド

Policy Center の概要	155
Policy Center (Standard Edition) との接続	156
ポリシー チェックとレポートの設定	157
ポリシー チェック結果の参照	161
ポリシーの設定	170

管理者

プロジェクトの作成と構成	180
Report Center 管理ページ	191
Project Center 管理ページ	213
その他の Report Center 設定	227
Cache Report Executor の設定	236

Report Center のツール	240
Team Server サンドボックスの作成	244
Data Collector から Team Server への DTP Engine 10.x レポートの転送	254
マルチキャスト DNS の使用	255
Microsoft Word へのエクスポートのカスタマイズ	256
ツールバーでのアプリケーションの無効化と有効化	257

Development Testing Platform の統合

Parasoft Test から Report Center/Project Center へのテスト結果の送信	259
CSV からの BTS データのインポート	260
バグ トラッキング システムおよび要求管理システムの統合	266
BTS/RMS Scanner と Updater	272
HP Quality Center との統合	286
Bugzilla との統合	294
IBM Rational ClearQuest との統合	297
Atlassian JIRA との統合	302
IBM Rational Change および Rational Synergy との統合	310
コード レビューの統合	315
Report Center と Emma の統合	318
Report Center とソース管理システムの統合	321
PTC Integrity ソース管理コントロール拡張との統合	325
サードパーティのツールから DTP へのテスト結果の送信	334
Microsoft Excel へのレポートのインポート	335
Java API	337

Report Center のインストール

Report Center の管理 (Windows)	338
Report Center の管理 (Linux)	340
使用できる Development Testing Platform 製品の確認	343
データの移行	344
MySQL サーバーのアップグレード	346
Configuration Manager の内部詳細 (Linux/Solaris)	347
Linux での初期データベースの作成とデータベースのアップグレード (コマンドライン メニューからの 実行)	349
crontab による自動起動の設定 (Linux)	351

トラブルシューティング

Cache Report Executor	353
Team Server	354
ソース表示機能	355
デバッグ ログ モードへの切り替え	358
索引	i

はじめに

このユーザー マニュアルは、Parasoft Development Testing Platform (DTP) を使用して Jtest、C++test、dotTEST などのバージョン 9.x 台のツールからテストおよび解析データを収集し、関連付け、レポートする組織向けです。

DTP を DTP Engine for Java、.NET、C/C++ と使用する場合のマニュアルについては、DTP の [ヘルプ] リンクからメインのユーザー マニュアルを参照してください。DTP Engine は、よりリッチなデータ、サードパーティ製アナライザー ツール実行の柔軟性、IDE、ビルド システム、その他のソフトウェア開発コンポーネントの強力なサポートを提供します。

Parasoft DTP について

ムであり、欠陥のあるソフトウェアに関するビジネス リスクの排除、ソフトウェア デリバリーの加速、継続的プロセス改善の推進に役立ちます。DTP は、静的解析、単体テスト、カバレッジ解析、実行時エラー検出などの品質プラクティスの適用状況をモニターして計測します。SDLC 全体で生成されたデータが収集され、関連付けと解析が行われ、インテリジェントで実行可能な指摘事項として提供されるため、コード変更の影響範囲に集中し、完全なコンプライアンス トレーサビリティを証明できます。

DTP の詳細については、DTP で [ヘルプ] リンクをクリックしてメインのマニュアルを参照してください。

Eclipse および Visual Studio との統合

Development Testing Platform は Eclipse および MS Visual Studio の IDE と統合します。Eclipse または MS Visual Studio のプラグインである Parasoft Test には、タスク アシスタントというコンポーネントがあります。このコンポーネントが Development Testing Platform のタスクを開発者の IDE に直接渡します。ユーザーは IDE (Eclipse または Visual Studio) を離れずに Development Testing Platform のタスクを検索することができます。

タスク アシスタントの使い方の詳細については Parasoft Test のドキュメントを参照してください。

ウィジェット

ウィジェットには、開発プロセスのさまざまな側面を簡潔に表す情報が表示されます。次の表は、Report Center で利用できるウィジェットのタイプの概要です。ライセンスによって、リスト中のすべてのウィジェットにアクセスできない場合があります (詳細については 739 ページの「全機能の DTP ライセンスへのアップグレード」を参照してください)。

タイプ	説明
ビルド結果ウィジェット	コードのビルド状況をモニターするのに役立つウィジェットです。7 ページの「ビルド結果ウィジェット」を参照。
コード ウィジェット	ファイルのリビジョン、コード行数、コード レビューのステータスなど、コードの統計情報およびアクティビティを表すウィジェットです。9 ページの「コード ウィジェット」を参照。
コンプライアンス ウィジェット	テスト対象アプリケーションのコーディング スタandardへの準拠に関する情報を表示するウィジェットです。Parasoft Static Analysis Execution Engine から DTP に送信された静的解析データを使用します。60 ページの「コンプライアンス ウィジェット」を参照。
欠陥ウィジェット	時間の経過とともに欠陥および改善の数が増加しているか減少しているかを表示するウィジェットです。11 ページの「欠陥 ウィジェット」を参照。
分析結果ウィジェット	コード解析の実行に関する情報をレポートするウィジェットです。69 ページの「分析結果ウィジェット」を参照。
メトリクス ウィジェット	DTP Engine が収集したメトリクス情報をレポートするウィジェットです。74 ページの「メトリクス ウィジェット」を参照。
ポリシー ウィジェット	組織の開発ポリシーに対する準拠および乖離に関する情報を表示するウィジェットです。77 ページの「ポリシー ウィジェット」を参照。
プロセス インテリジェンス ウィジェット	DTP 5.2 から、プロセス インテリジェンス ウィジェットは Parasoft Marketplace からダウンロードするか、Process Intelligence Engine (PIE) Slice Designer を使ってインポートする必要があります。詳細については Parasoft 製品の担当者にお問い合わせください。
Project Center ウィジェット	スクラムなどの反復型開発プロセスを採用するチームが、完了した作業を計測するのに役立つウィジェットです。13 ページの「Project Center ウィジェット」を参照。
静的解析 (9.x) ウィジェット	バージョン 9.x の Parasoft Test 製品 (C++test 9.x、Jtest 9.x、dotTEST 9.x、および SOAtest 9.x) から DTP に送信された静的解析データを返すウィジェットです。15 ページの「静的解析 (9.x) ウィジェット」を参照。

タイプ	説明
静的解析ウィジェット	Parasoft DTP Engine から DTP に送信された静的解析データを返すウィジェットです。84 ページの「静的解析ウィジェット」を参照。
テスト (9.x) ウィジェット	バージョン 9.x の Parasoft Test 製品 (C++test 9.x、Jtest 9.x、dotTEST 9.x、および SOAtest 9.x) から DTP に送信されたテスト実行データを返すウィジェットです。19 ページの「テスト (9.x) ウィジェット」を参照。
テスト ウィジェット	単体テスト、テスト カバレッジ、機能テスト、手動テストなどのテスト実行結果を表示するウィジェットです。102 ページの「テスト ウィジェット」を参照。
カスタム ウィジェット	外部リソースをダッシュボードに埋め込んだり、独自のウィジェット HTML を記述したり、Parasoft マーケットプレイスから入手したウィジェットをデプロイすることによって、DTP を拡張するためのカスタム ウィジェット コンテナです。110 ページの「カスタム ウィジェット」を参照。

以降の表は、Report Center で利用できるウィジェットのタイプの説明と設定オプションです。ウィジェットの ID は、ダッシュボード テンプレートで使用されます。ウィジェット ID の使用方法については、688 ページの「カスタム ダッシュボード テンプレート」を参照してください。

ビルド結果ウィジェット

ビルド結果

ビルド中に失敗したファイル、警告のあるファイル、および成功したファイルの数を 9.x バージョンの Parasoft ツールによってレポートされたとおりに表示します。

アクション

ウィジェットをクリックすると、[ビルド結果] レポートを参照できます。詳細については 62 ページの「[ビルド結果] レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Build Results",  
"type": "legacy",  
"id": "BuildsHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

Jenkins ジョブ結果

Jenkins ジョブのステータスおよびビルド結果を表示します。Jenkins ベースのウィジェットが正常に機能するためには、Jenkins で匿名ユーザーの読み取りアクセスが許可されていなければなりません。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Jenkins Job Result",  
"type": "legacy",  
"id": "JenkinsJob"
```

設定

- **タイトル** : デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **Jenkins サーバー** : Jenkins サーバーの URL です。
- **Jenkins ジョブ** : ジョブ名です。

コードウィジェット

チェックイン

関連付けられたプロジェクトに関して、バージョン 9.x の Parasoft ツールによってソース管理システムにコミットされたファイル リビジョンの数を表示します。

アクション

ウィジェットをクリックすると、[ソース コード チェックイン] レポートを参照できます。詳細については 59 ページの「[ソース コードのチェックイン] レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Check-ins",  
"type": "legacy",  
"id": "RevisionHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

コード ベースのサイズ

プロジェクトのコード行数を表示します。

アクション

ウィジェットをクリックすると、[コード ベースのサイズ] レポートを参照できます。詳細については 125 ページの「コード ベースのサイズ」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Code Base Size",
"type": "legacy",
"id": "CodeBaseSizeHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウンメニューから期間を選択します。

コード レビュー

バージョン 9.x の Parasoft ツールによって支援されたコード レビュー プロセスの全般的なステータスを表示します。

アクション

ウィジェットをクリックすると、[コード レビュー アクティビティ] レポートを参照できます。詳細については 68 ページの「コード レビュー アクティビティ」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Code Review",
"type": "legacy",
"id": "CodeReviewSummaryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウンメニューから期間を選択します。

欠陥ウィジェット

欠陥の傾向

時間の経過に伴う欠陥数の変化を表示します。

アクション

ウィジェットをクリックすると、[欠陥]レポートを参照できます。詳細については 296 ページの「欠陥および改善レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Defect Trend",  
"type": "legacy",  
"id": "PrHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

改善の傾向

時間の経過に伴う改善数の変化を表示します。

アクション

ウィジェットをクリックすると、[改善]レポートを参照できます。詳細については 296 ページの「欠陥および改善レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Enhancement Trend",
```

```
"type": "legacy",
"id": "FrHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウンメニューから期間を選択します。

週ごとの新しい欠陥

レポートされた欠陥の数を週ごとに棒グラフで表示します。欠陥はステータスによって色分けされます。

- 赤：新規または再オープン
- 黄：割り当て済み
- 黄緑：解決済み
- 深緑：検証済み

アクション

棒グラフの色分けされたセクションの上にマウスポインタを置くと、その週の欠陥数が表示されます。グラフの棒をクリックすると、Project Center で欠陥のページを参照できます。詳細については 418 ページの「欠陥 / 改善の編集」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "New Defects by Week ",
"type": "legacy",
"id": "DefectsLastWeeks"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。

Project Center ウィジェット

イテレーションのバーンダウン

選択されているプロジェクトで開かれているすべてのイテレーションに関して、理想的な完了率に対する完了した作業の割合を表示します。

アクション

ウィジェットをクリックすると、Project Center でイテレーションが開かれます。詳細については 429 ページの「イテレーションへのプロジェクトの分割」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Iteration Burndown",  
"type": "legacy",  
"id": "ProjectIterationsBurndown"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **ゲート**：ドロップダウン メニューから品質ゲートを選択します。プロジェクトに品質ゲートがない場合、利用できるデータがないことが表示されます。

要求バーンダウン

時間の経過に伴う要求のストーリー ポイントの変化を表示します (バーンダウン)。

アクション

ウィジェットをクリックすると、Project Center で要求が開かれます。詳細については 396 ページの「要求の作成と管理」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。


```
"name": "Requirements Burndown",
"type": "legacy",
"id": "RequirementsBurndownChartWidget"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **要求 ID**：Project Center の要求 ID を入力します (429 ページの「イテレーションへのプロジェクトの分割」を参照)。
- **開始日**：イテレーションの開始日を入力します。
- **終了日**：イテレーションの終了日を入力します (任意)。

タスク

イテレーションのすべてのオープン中のタスクを表示します。

アクション

イテレーション ID をクリックすると、Project Center でイテレーションが開かれます。詳細については 429 ページの「イテレーションへのプロジェクトの分割」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Tasks",
"type": "legacy",
"id": "TasksInProgress"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **プロジェクト**：プロジェクトを選択します。
- **イテレーション**：イテレーションを選択します。
 - クローズされたイテレーションも含めるには、[クローズされたものも表示] オプションを有効にします。
- **ステータス**：タスクのステータスでウィジェットの表示結果をフィルタリングします。
- **テスト ステータス**：テストのステータスでウィジェットの表示結果をフィルタリングします。

静的解析 (9.x) ウィジェット

バージョン 9.x の Parasoft Test 製品 (C++test 9.x、Jtest 9.x、dotTEST 9.x、および SOAtest 9.x) から DTP に送信された静的解析データを返すウィジェットです。

カテゴリ別エラー

バージョン 9.x の Parasoft Test 製品から送信された静的解析および単体テストの結果をカテゴリ別に表示します。

アクション

グラフの棒をクリックすると、[カテゴリ別エラー] レポートが開きます。詳細については 85 ページの「[カテゴリ別エラー] レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Errors by Category",  
"type": "legacy",  
"id": "ErrorsByCategoryHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

重要度別エラー

バージョン 9.x の Parasoft Test 製品から送信された静的解析および単体テストの結果を重要度別に表示します。

アクション

グラフの棒をクリックすると、[重要度別エラー] レポートが開きます。詳細については 94 ページの「[重要度別エラー] レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Errors by Severity",  
"type": "legacy",  
"id": "ErrorsBySeverityHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
 - **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
 - **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。
-

カテゴリ別 最近のエラー

最後のテスト実行中に見つかったエラーの詳細をカテゴリ別に表示します。バージョン 9.x の Parasoft Test 製品から送信された結果を表示します。

アクション

[エラー] 列のリンクをクリックすると、選択されたカテゴリのエラーの詳細レポートが開きます。詳細については 85 ページの「[カテゴリ別エラー] レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Most Recent Errors by Category",  
"type": "legacy",  
"id": "ErrorsByCategorySeverityDetailsGrid"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

重要度別 最近のエラー

最後のテスト実行中に見つかったエラーの詳細を重要度別に表示します。バージョン 9.x の Parasoft Test 製品から送信された結果を表示します。

アクション

ウィジェットをクリックすると、[重要度別エラー] レポートが開きます。詳細については 94 ページの「[重要度別エラー] レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Most Recent Errors by Severity",  
"type": "legacy",  
"id": "ErrorsBySeverityCategoryDetailsGrid"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

静的解析 - ファイル

成功した静的解析テストおよび失敗した静的解析テストのファイル数を表示します。バージョン 9.x の Parasoft Test 製品から送信された結果を表示します。

アクション

ウィジェットをクリックすると、[静的解析 - ファイル] レポートが開きます。詳細については 197 ページの「[静的解析 - ファイル] レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Static Analysis - Files",  
"type": "legacy",
```

```
"id": "StaticFilesHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウンメニューから期間を選択します。

静的解析 - 違反

プロジェクト内で検出された違反の数を表示します。バージョン 9.x の Parasoft Test 製品から送信された結果を表示します。

アクション

ウィジェットをクリックすると、[静的解析 - 違反] レポートが開きます。詳細については 187 ページの「[静的解析 - 違反] レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Static Analysis - Violations",  
"type": "legacy",  
"id": "StaticViolationsHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウンメニューから期間を選択します。

テスト (9.x) ウィジェット

カバレッジ

選択されたプロジェクトの単体テスト カバレッジを表示します。

アクション

ウィジェットをクリックすると、[カバレッジ] レポートが開きます。詳細については 151 ページの「カバレッジ レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Coverage",  
"type": "legacy",  
"id": "CoverageHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
 - **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
 - **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。
-

機能テスト統計

最後に実行された機能テストの成功したテストのパーセンテージ、テスト数、失敗の数、未完了のテストの数、実行時間の合計を表示します。また、選択された日付範囲内に各統計値に変化があったかどうかを示します。

アクション

ウィジェットをクリックすると、[タイプ別テスト] レポートが開きます。詳細については 286 ページの「タイプ別テスト」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Functional Tests - Statistics",  
"type": "legacy",  
"id": "functionalTestMetricWidget"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

機能テスト サマリー

選択された日付範囲内の成功したテストのパーセンテージと、成功率の変化を表示します。

アクション

ウィジェットをクリックすると、[タイプ別テスト] レポートが開きます。詳細については 286 ページの「タイプ別テスト」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Functional Tests - Summary",  
"type": "legacy",  
"id": "FunctionalTestSummary"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

手動テスト セッション

選択されたプロジェクトに対して実行または計画されたすべての手動テスト セッションのリストを表示します。

アクション

ウィジェットをクリックすると、[手動テスト セッション] レポートが開きます。詳細については 149 ページの「手動テスト セッション」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Manual Test Sessions",  
"type": "legacy",  
"id": "TestingSessionsRunsHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

テスト (ファイル)

未完了のテスト ケース、失敗したテスト ケース、および成功したテスト ケースの数を表示します。

アクション

ウィジェットをクリックすると、[テスト (ファイル)] レポートが開きます。詳細については 285 ページの「テスト (ファイル)」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Tests (Files)",  
"type": "legacy",  
"id": "TestHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウンメニューから期間を選択します。

単体テスト

時間の経過に伴う単体テストの失敗の数およびテスト ケース数の合計の変化を表示します。

アクション

ウィジェットをクリックすると、[テスト ケース] レポートが開きます。詳細については 317 ページの「テスト ケース レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Unit Tests",  
"type": "legacy",  
"id": "UnitTestingHistoryChart"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウンメニューから期間を選択します。

単体テスト統計

最後に実行された単体テストの成功したテストのパーセンテージ、テスト数、失敗の数、未完了のテストの数、実行時間の合計を表示します。また、選択された日付範囲内に各統計値に変化があったかどうかを示します。

アクション

ウィジェットをクリックすると、テスト ケース レポートが開きます。詳細については 317 ページの「テスト ケース レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Unit Tests - Statistics",  
"type": "legacy",  
"id": "unitTestMetricWidget"
```

設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。
- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウン メニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウン メニューから期間を選択します。

単体テスト サマリー

選択された日付範囲内の成功した単体テストのパーセンテージと、成功率の変化を表示します。

アクション

ウィジェットをクリックすると、[テスト ケース] レポートが開きます。詳細については 317 ページの「テスト ケース レポート」を参照してください。

カスタム ダッシュボード プロパティ

このウィジェットをカスタム ダッシュボードに追加するには、次のプロパティを使用します。

```
"name": "Unit Tests - Summary",  
"type": "legacy",  
"id": "UnitTestSummary"
```

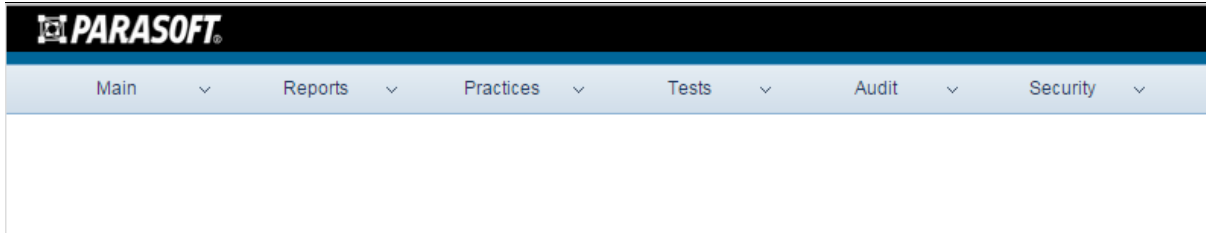
設定

- **タイトル**：デフォルトのタイトルの代わりにダッシュボードに表示される新しいタイトルを入力します。

- **フィルター**：ダッシュボード設定を選択してダッシュボードのフィルターを使用するか、ドロップダウンメニューからフィルターを選択します。
- **期間**：ダッシュボード設定を選択してダッシュボードの日付範囲を使用するか、ドロップダウンメニューから期間を選択します。

Parasoft Test ツールのレポート (9.x 以前)

Parasoft Test ツール (C/C++test、Jtest、dotTEST、SOAtest) のバージョン 9.x 以前に関連するウィジェットは、ソフトウェア品質アクティビティの詳細情報を表示するネストしたレポートにリンクします (15 ページの「静的解析 (9.x) ウィジェット」および 19 ページの「テスト (9.x) ウィジェット」を参照してください)。また、ブラウザのアドレス バーに [DTP_DOMAIN] /grs/reports.jsp と入力すると、レポート メニューにアクセスすることができます (レポート メニューは Concerto および DTP 5.1 以前にあった機能です)。



レポートのブックマーク、レポート範囲の変更、表のソート、データの表示方法の変更を行うことができ、開発プロセスのデータについて詳しく知ることができます。

レポートのブックマーク

[Bookmark] リンクをクリックして URL を変更すると、ブラウザでレポートをブックマークすることができます。



[Bookmark] リンクは、ブラウザのブックマーク フォルダーにレポートへのリンクを追加するわけではありません。[Bookmark] リンクをクリックしたあとに、ブラウザのブックマーク機能を使って URL を追加する必要があります。

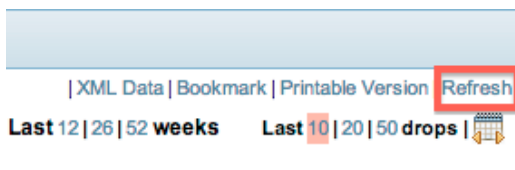
ブックマークされたレポートを表示するには、Report Center にログインする必要があります。Report Center にログインしていない状態で、ブックマークされたレポート ページをブラウザから選択した場合、ログインするよう促されます。ログインすると、ブックマークされたレポートが開き、現在のデータが表示されます。

また、データを XML としてエクスポートし、別のプログラムにインポートすることもできます。詳細については 335 ページの「Microsoft Excel へのレポートのインポート」を参照してください。

レポートのリフレッシュ

レポートを参照するとき、DTP データベースからのデータが DTP サーバーにキャッシュされます。別のユーザーが同じパラメーター (プロジェクトや期間など) を使って同じレポートを参照する場合、データは DTP データベースからではなくキャッシュから取得されます。

デフォルトでは、1日に1回00:00時に、DTPサーバー上のすべてのレポート キャッシュが消去されます。ただし、必要なときに DTP データベースの最新のデータでレポートをリフレッシュすることができます。レポートをリフレッシュするには、[Refresh] リンクをクリックします。



ダッシュボード ビューからアクセスされるレポートを含め、すべての DTP レポートにはタイムスタンプがあります。そのため、作業しているデータの新鮮さ / 古さを知ることができます。

4	19326	19330
0	19341	19341
0	19341	19341
49	19292	19341
0	19353	19353
2	19390	19392
37	19408	19445
560	18849	19409

1 - 30 / 30 items

Report generated: 2014-05-21 9:21:27 am

レポートのフィルタリング

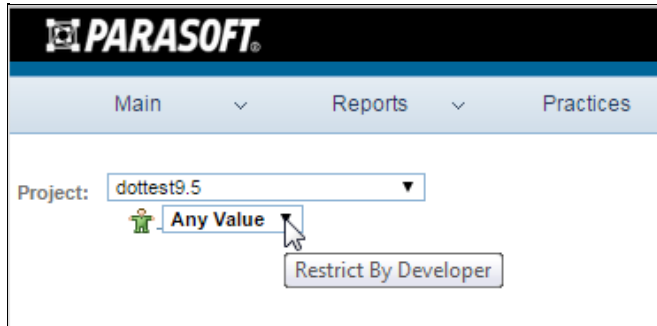
データの性質によって、日付、期間、プロジェクト、あるいはプロジェクト チーム メンバーでレポートをフィルタリングできる場合があります。以下のフィルター オプションを利用できます。

プロジェクトによるフィルタリング

[Project] ドロップダウン メニューをクリックし、プロジェクトを選択します。

プロジェクトとチームメンバーによるフィルタリング

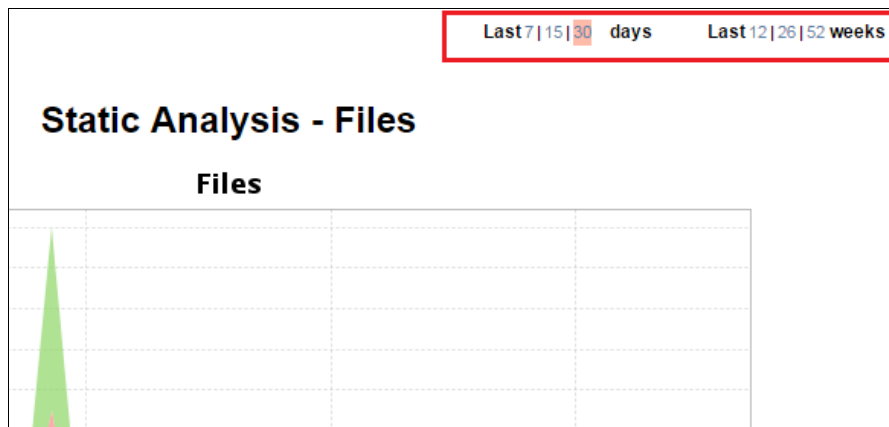
1. [Project] ドロップダウンメニューからプロジェクトを選択します。
2. [Restrict by Developer] ドロップダウンメニューからチームメンバーを選択します。



期間によるフィルタリング

最近の日数または週数でデータをフィルタリングすることができます。

- **7**、**15**、または **30** をクリックすると、最近の日数でデータがフィルタリングされます。
- **12**、**26**、または **52** をクリックすると、最近の週数でデータがフィルタリングされます。



最新のドロップによるフィルタリング

10、20、または 50 をクリックすると、最新のドロップ数でデータをフィルタリングすることができます。

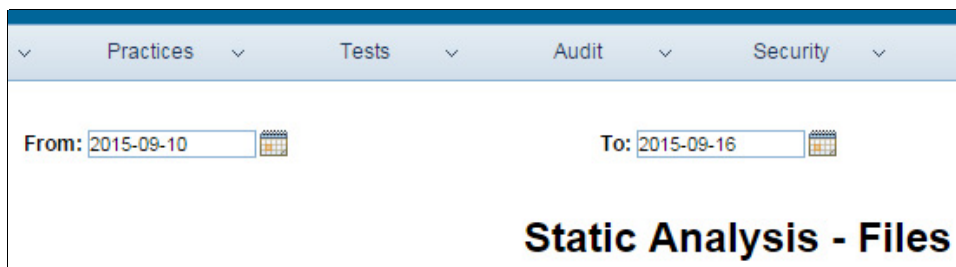


データ 範囲によるによるフィルタリング

1. カレンダー アイコンをクリックしてデータ範囲モードを切り替えます。



2. [From] および [To] フィールドに日付を入力して、データ範囲を指定します。



表のソート

列ヘッダーをクリックすると、表をソートすることができます。

Date ▼	Incomplete Tests	Failed Tests	Passed Tests	Tool	User	Total Tests
Mar 30,06 (419856)	0	0	1	Source Scanner	builds	1
Mar 29,06 (419897)	0	3	0	JUnit	sdtest	3
Mar 29,06 (419896)	0	3	7	JUnit	sdtest	10
Mar 29,06 (419895)	0	2	1	JUnit	sdtest	3
Mar 29,06 (419893)	0	0	54	RTest	devtest	54
Mar 29,06 (419891)	0	15	85	RTest	devtest	80
Mar 29,06 (419890)	0	0	79	RTest	devtest	79
Mar 29,06 (419886)	0	0	79	RTest	devtest	79
Mar 29,06 (419888)	0	5	75	RTest	devtest	80
Mar 29,06 (419884)	0	3	0	JUnit	sdtest	3

ユーザーのスタートアップレポート

バージョン 9.x 以前のレポートにユーザーがアクセスしたときにデフォルトで表示するレポートを指定できます。[Main] > [Startup Report] を選択してデフォルト レポートを有効にします。

Main	Reports	Practices	Tests	Audit	Security
User Startup Report					
Select your startup report:					
Reports	Practices	Tests	Audit	Security	
Policy Report <input type="checkbox"/>	Static Analysis <input type="checkbox"/>	Errors By Category <input type="checkbox"/>	Metric Top Results <input type="checkbox"/>	Security Tests <input type="checkbox"/>	
Project Portfolio <input type="checkbox"/>	Unit Testing <input type="checkbox"/>	Errors By Severity <input type="checkbox"/>	Single Metric Overview <input type="checkbox"/>	Security Violations <input type="checkbox"/>	
	Builds <input type="checkbox"/>	Tests By Date <input type="checkbox"/>			
	Code Review Activity <input type="checkbox"/>	Tests Overview <input checked="" type="checkbox"/>			
	Code Review Status <input type="checkbox"/>	Change-Based Testing			
	Source Code Check-ins <input type="checkbox"/>	Requirements/Defects <input type="checkbox"/>			
	Tools Usage <input type="checkbox"/>	Test Scenarios <input type="checkbox"/>			
		Manual Test Sessions <input type="checkbox"/>			
		Recent Test Logs <input type="checkbox"/>			

レポートの参照

レポートは、一目で把握しやすい量のデータを表示します。多くのレポートは、クリックすることで、特定のセグメントに関するより詳細な情報を参照できます。

スケジュール、コスト、品質の管理

バージョン 9.x 以下の Parasoft のコード解析およびテスト ツールから Parasoft DTP (または Concerto) に送られるデータは、レポートを中心としたパラダイムに従って、開発プロセスを知る上で重大な手がかりをユーザーに与えます。この章では、リリース時期の決定に役立つ、レポート中心のパラダイムを利用するための指針について説明します。

このセクションでは、スケジュール、費用、および品質の管理について説明します。

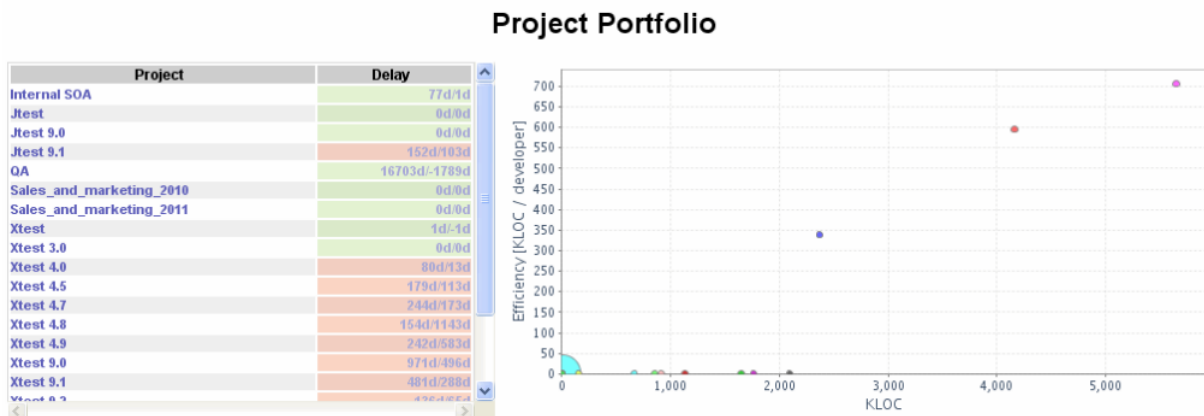
このセクションの内容：

- プロジェクトがスケジュールどおりかの検証
- 毎日変更されるソース コード量の検証
- コードが徹底的にテストされているかの検証
- バグが減少し、コントロールされているかの検証
- 脆弱な要求の特定とプロジェクト リリースの決定

プロジェクトがスケジュールどおりかの検証

Report Center のプロジェクト ポートフォリオは、開発チームを超えてすべてのプロジェクトを 1 つのビューで提供します。プロジェクトを超えてデータを参照できるため、開発メトリクスがプロジェクトの問題の可能性を示したときにユーザーはすぐに気づくことができます。

このレポートを表示するには、レポート ビューで [レポート] > [プロジェクト ポートフォリオ] をクリックします。



全体的なプロジェクト スコアに加えて、Report Center は傾向、差異、到着予想時刻 (ETA)、および設定されたマイルストーンに達する予測日を示すほか、次のプロジェクト要素ごとにスコアを示します。

- PR
- FR
- テスト (手動テスト以外のすべてのテスト)
- 単体テスト (UT)
- コーディング スタンダード (CS)

- 効率性

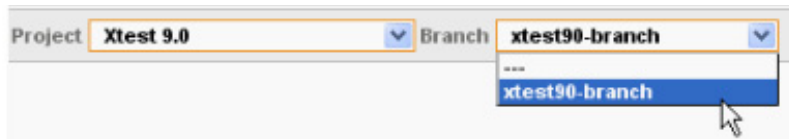
これらのデータからさらに詳細なデータを参照することで、プロジェクト ステータスを試算し解析することができます。

毎日変更されるソース コード量の検証

プロジェクトが進行するにつれて、作業が進捗しているかを把握したい場合があります。ソース管理システムと統合すること、コード ウィジェットを使用してチームの生産性を把握できます (9 ページの「コード ウィジェット」を参照)。

Report Center とソース管理システムの統合の詳細については、321 ページの「Report Center とソース管理システムの統合」を参照してください。

ブランチ フィルターがあるプロジェクトで作業している場合、[ブランチ] ドロップダウン リストからブランチ名を選択して、表示するコード統計を特定のブランチだけに制限することができます。(ブランチ フィルターを設定するには、[管理] から [プロジェクトの編集] ページを開き、[プロジェクト フィルターの定義] の [ソース管理フィルター] を使用します)。

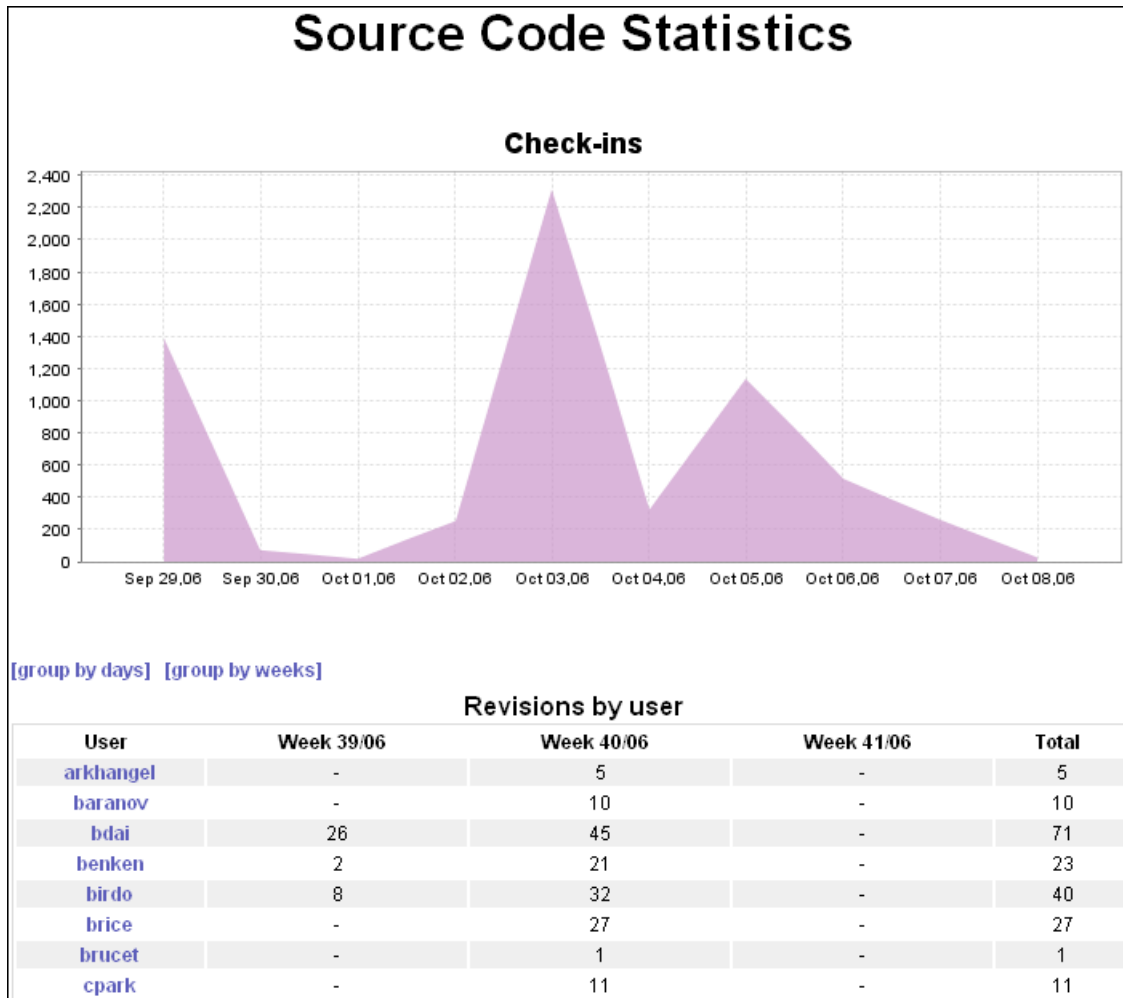


[コード ベースのサイズ] グラフからさらに詳細なデータを参照して、以下の疑問に対する答えを得ることができます。

- 誰がコードに対して作業しているのか？
- 誰がどのコードをチェックインしているのか？
- 各チーム メンバーは実際に何に対して作業しているのか？

グラフは、コード ベースの変更の特徴について包括的なビューを提供します。全般的なコードの成長をプロジェクト、チーム、チーム メンバーごとにグラフ化します。Report Center のダッシュボードは、1 日または 1 週間あたりの新しいファイル リビジョンの数、1 ファイルあたりの変更された行数、ファイル変更の作成者といった重要な情報をグラフ化します。

レポートされるコードの成長の統計は、開発の効率性 (または非効率性) を客観的に測ったり開発の進捗を記録するための手段を提供するほか、プロジェクトのゴールまたは締切が危機にあることをメトリクスが示したときに積極的に対応するための手段を提供します。コードの成長のデータと他の Report Center レポートで提供される他の開発メトリクス (フィーチャー リクエスト、PR、テスト結果など) を関連付けることで、コードの品質と準備状況に対するコードの変更の影響を定期的に継続して評価することができます。

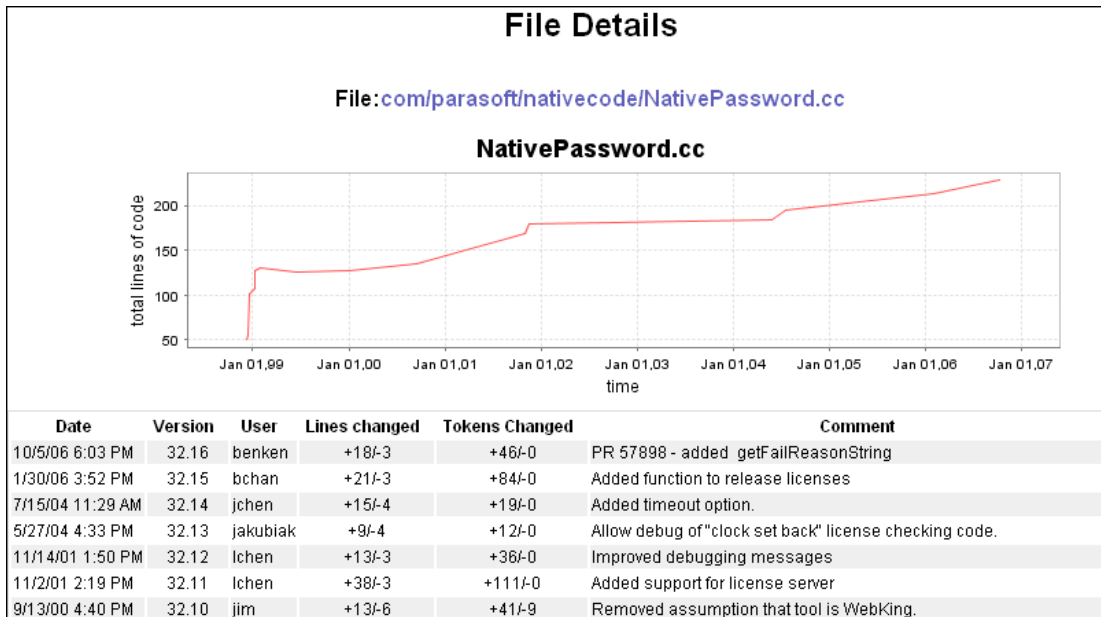


[ソースコードのチェックイン]のグラフから、さらにソースファイルとファイルのすべての変更履歴を参照できます。プロジェクトは、各チームメンバーがどのソースコードに対して作業しているかが分かるように編成できます。このグラフは、アーキテクトに開発の生産性に関する情報を提供します。ユーザーは [ファイルの詳細] ページに進み、特定の機能を構成する特定のファイルの成長を参照して、その成長が優先順位レベルに対して適切かどうかを評価することができます。

ファイルの詳細

1. コードベースのサイズグラフをクリックしてソース管理サマリーレポートを開きます。

2. [ファイル] 列のリンクをクリックすると、ファイルの詳細レポートが表示されます。



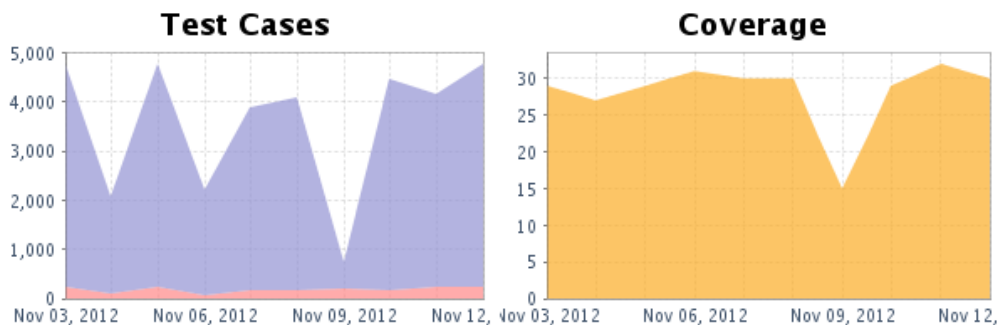
[コードベースのサイズ] グラフとそこからアクセスできるグラフと表から、ユーザーはソース管理システムのアクティビティを見ることができます。誰が何に対して作業しているかを見て、各プロジェクト、機能、およびファイルの進捗を評価することができます。

コードが徹底的にテストされているかの検証

コードが徹底的にテストされたかを検証するために、単体テストデータ、タイプ別テストグラフに表示されるデータの結果、および手動テストグラフのデータを調べることができます。

[プラクティス] > [単体テスト] を選択して、テストケースとカバレッジのグラフおよびその詳細レポートにアクセスします。

Unit Testing



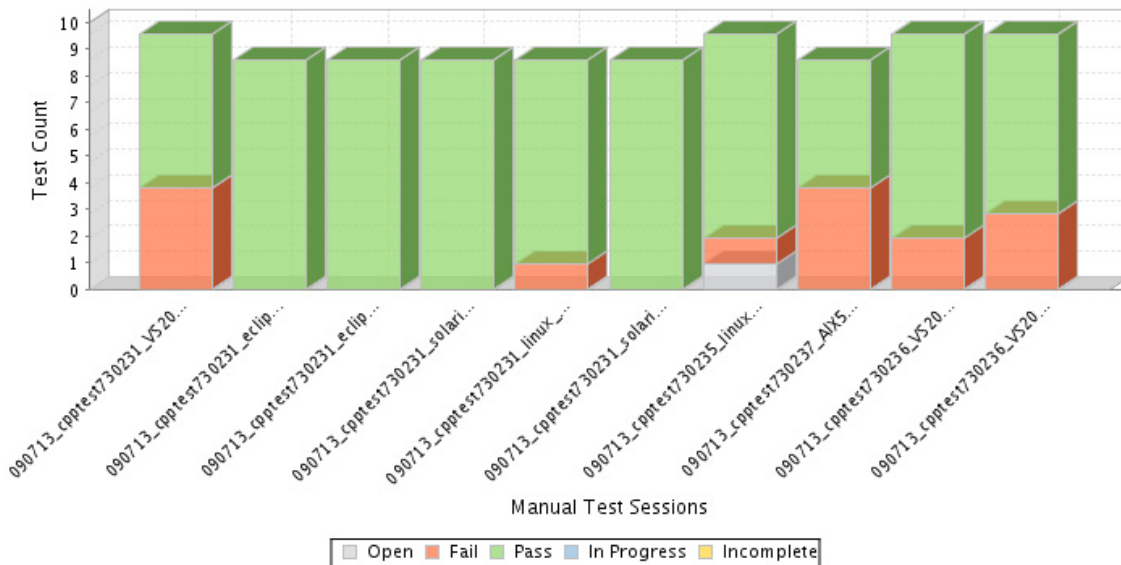
自動化された単体テストでは、ユーザーはどの程度テストが実行されたのか、そしてそのテスト結果を見ることができます。テストは成功しているがテスト カバレッジが低すぎる場合、コードは堅牢で機能的に安全であるが、品質の完全さを検証するテストが不十分な可能性があります。

カバレッジ レポートは、どのテスト ケースが成功したか、何個のテスト ケースが成功したか、および開発プロセスのさまざまな段階で実行されたテスト コードの割合を示します。

単体テスト レポートを参照した後、[テスト] グラフを開いてコーディング スタandardに準拠しているかどうかを確認できます ([テスト] グラフには他のテスト タイプも表示されます)。[タイプ別テスト] レポートには、特定の日における未完了、失敗、成功のコーディング スタandard テストの数が表示されます。また、テストが実行されたマシンとツールもリストされます。

手動テストの徹底度を検証するには、[手動テスト セッション] グラフ (図 1) に移動します。このグラフからさらに [手動テスト セッション] レポートのグラフと表に進むと、コード カバレッジの詳細に加えて、項目、ユース ケース、保留中のテスト、実行されたテストのそれぞれの数と各モジュールにかかったテスト時間を参照できます。各モジュールのテスト結果 (進行中、未完了、失敗、成功) も見ることができます。

図 1: 手動テスト セッション

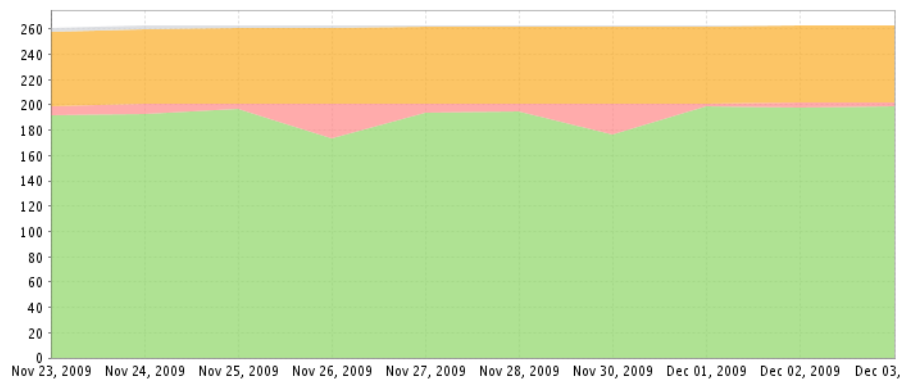


手動テストの詳細については、149 ページの「手動テスト セッション」を参照してください。

バグが減少し、コントロールされているかの検証

[欠陥の傾向] ウィジェットをクリックして欠陥レポートにアクセスすると、アプリケーション中のバグの可能性が表示されます (11 ページの「欠陥ウィジェット」を参照)。欠陥のレポートと修正のプロセスは、機能の場合と同じです。

Defects



脆弱な要求の特定とプロジェクトリリースの決定

機能と要求が実装されるにつれ、テストが発生し、バグが発見されてレポートされます。バグに関して次の質問に回答する必要があります。

- バグの数は減少しているか？
- バグはコントロールされているか？

バグがコントロールされている場合、Bugs History Overview グラフにおいて、修正されたバグの数を表す線は繰り返し上下します。これは、バグが発見されて修正され、さらにバグが発見されて修正されるという状況の繰り返しを反映しています。このグラフ線はのこぎりの歯のようなジグザグのパターンです。小さなジグザグのパターンは問題ありません。重要なのは、一定の割合でバグが発見されて修正されているかどうかです。

では、ジグザグのパターンの繰り返しがない場合はどうでしょうか？ グラフ線は着実に上昇していて、下降する部分がまったくありません。これは、バグがどんどん発見されているがほとんど解決されていないことを意味します。そしてさらに、調査が必要な問題があることを意味します。

また、バグがほとんど発見されない、あるいはまったく発見されない場合はどうでしょうか？ そのような状況は、テストが発生していないか、開発者が完璧なコードを作成していることを表します。いずれにせよ、グラフ線がジグザグのパターンではない場合は、何らかの調査を実施して行動を起こすべきです。

問題の核心を突き止めるには、バグの修正とソースコードを相関できるシステムを用意するのが最善の方法です。このシステムは、PR 情報と FR 情報をコードに関連して記録するために使用できる IDE プラグインとして実装することができます。これは、どのファイルが修正され、特定の要求のために変更されたかを識別するのに役立ちます。たとえば、最近のプロジェクトに対して手動テストを実施した場合、手動テストの結果を参照して次のような質問に回答し、問題の原因を特定することができます。

- 誰がテストを実施したのか？
- どのくらいの時間をテストにかけたのか？
- テストの内容は何か？
- テストの結果はどうか？

問題を発見したら、問題に関連する機能をリリースに含めるかどうか、あるいはリリース日を確定できるよう、次のリリースまで機能の実装を延期するかどうかを決定する必要があります。この決定に役立つのは、バグを修正するために追加が必要なコード量を把握することです。必要なコード量が多ければ多いほど、コストは上昇します。

違反のコストの算定

Report Center には [OWASP Top 10 - ピラミッド] ウィジェットがあり、欠陥に関連するコストを理解するのに役立ちます。

Category	Cost	Total
A1 - Injection	89	89
A2 - Broken Authentication and Session Management	288	377
A3 - Cross-Site Scripting (XSS)	7	384
A4 - Insecure Direct Object References	0	384
A5 - Security Misconfiguration	0	384
A6 - Sensitive Data Exposure	292	676
A7 - Missing Function Level Access Control	0	676
A8 - Cross-Site Request Forgery (CSRF)	7	683
A9 - Using Components with Known Vulnerabilities	0	683
A10 - Unvalidated Redirects and Forwards	0	683

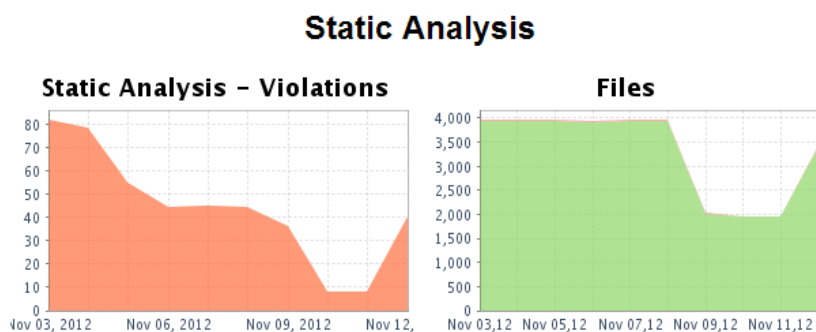
違反のコストは、ルールの違反数にそのルールの重みを掛けることによって計算されます。デフォルトでは、重みは 1 です。ルールの重みを記述した .xlsx ファイルを変更し、違反のコストを算定する方法をカスタマイズできます。この .xlsx ファイルは次のディレクトリにあります。

DTP_HOME/grs/datasource/JavaOWASPTop10RulesWeight.xlsx.

静的解析レポート

静的解析レポートは、チームに静的解析を導入し、静的解析がソフトウェア開発ライフサイクルの一部として長く使用されていくことを助ける手段として機能します。[プラクティス] メニューの [静的解析] をクリックすると、次の 2 つのグラフが表示されます。

- [静的解析 - 違反] レポート

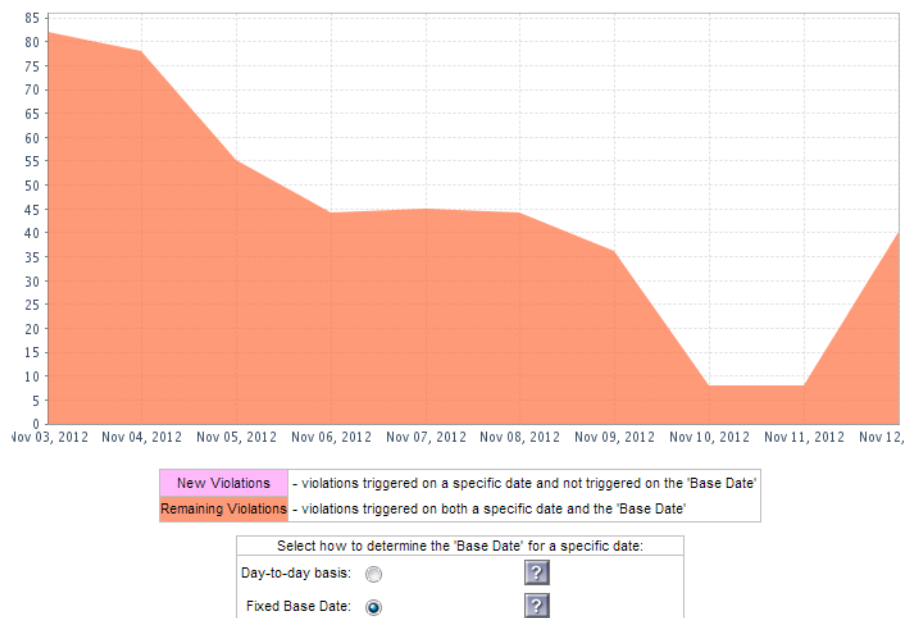


この画面を表示するには：[プラクティス](#) > [静的解析](#)

[静的解析 - 違反] レポート

[静的解析 - 違反] レポートを参照すると、開発者はプロジェクト コードで検出された静的解析違反の数をすばやく把握できます。そしてこのレポートからさらに、違反のソース コードや担当した開発者についての情報にアクセスすることができます。[静的解析 - 違反] レポートには、選択された

プロジェクトで検出された静的解析違反の傾向が表示されます。違反の総数や、最後に解析を実行した後に入り込んだ新規違反を簡単に参照できます。



Date	New Violations*	Fixed Violations*	Remaining Violations*	Suppressed violations	Violations	Files failed
v 12, 2012	0	42	40	2054	40	29
v 11, 2012	0	74	8	3007	8	3
v 10, 2012	0	74	8	3007	8	3
v 09, 2012	0	46	36	721	36	27
v 08, 2012	0	38	44	3728	44	30
v 07, 2012	0	37	45	3728	45	31
v 06, 2012	0	38	44	3713	44	30
v 05, 2012	0	27	55	3723	55	36
v 04, 2012	0	4	78	3723	78	37

この画面を表示するには：プラクティス > 静的解析 > 静的解析 - 違反

[静的解析 - 違反] レポート下部の表には次の情報が表示されます。

- **新規違反** 前回の解析実行後に検出された違反の数。
注意：[固定のベース日] に切り替えると、指定された日付から後に検出された違反を参照できます。
- **修正済み違反** 前回の解析実行後に修正された違反の数。
- **残りの違反** 前回の解析実行後に修正されていない違反の数。
- **抑制された違反** ソース コードで抑制された静的解析の数。
- **違反** ソース コードで検出された違反の総数。
- **失敗したファイル** 違反が検出されたファイルの数。

ドロップ日付ごとに、選択されたプロジェクトのコード内で見つかった静的解析違反の正確な数が表示されます。レポートは非デルタ モードに切り替えることができます (下記の注意を参照) 。

このレポートには 2 つのモードがあります。

- デルタ モードは、上記で説明しているデフォルトのビューであり、新規違反と残りの違反を表示します。
- 非デルタ モードは、日ごとの違反数と抑制された違反数を表示します。

モードを切り替えるには、[... に切り替え] リンクをクリックします。

[静的解析 - 違反] ページ

[静的解析 - 違反] には 3 種類の表示があります。以下の 3 種類の方法でデータを表示できます。

- 静的解析の違反 - タイプ別：Static Analysis Violations グラフの下にある表で、[違反] 列の違反数をクリックすると、このビューが表示されます。
- 静的解析の違反 - ファイル別：Static Analysis Violations グラフの下にある表で、[失敗したファイル] 列のファイル数をクリックすると、このビューが表示されます。
- 静的解析の違反 - 開発者別：タイプ別違反またはファイル別違反ビューを表示し、[開発者] 列のヘッダーをクリックすると、このビューが表示されます。

静的解析の違反 - タイプ別

[静的解析の違反 - タイプ別] グラフの棒は、レポートされた違反数が多い違反カテゴリを表しています。上位 10 個までのカテゴリが表示されます。



この画面を表示するには: プラクティス > 静的解析 > 静的解析 - 違反 > [違反] の数 (表の中)

詳細の表には、グラフ中の各違反タイプについて次の情報が表示されます。

- **ファイル**: 違反があるファイルの名前。
- **違反**: そのファイルで検出された違反の数。
- **開発者**: その違反に責任がある開発者のログイン ID。

表のリンクをクリックするとビューが変化し、さらに詳しい情報を参照できます。

- [ファイル] 列のヘッダー: ファイル別ビューに切り替わります。
- [開発者] 列のヘッダー: 開発者別ビューに切り替わります。
- 違反タイプ名: [静的解析の違反 - 違反タイプ別] ページが表示されます。詳細については 44 ページ を参照してください。

- ファイル名：[静的解析の違反 - ファイル別] ページが表示されます。詳細については 45 ページ を参照してください。
- 開発者名：[静的解析の違反 - 開発者別] ページが表示されます。詳細については 45 ページ を参照してください。

静的解析の違反 - ファイル別

[静的解析の違反 - ファイル別] グラフの棒は、レポートされた違反数が多いファイルを表しています。上位 10 個までのファイルが表示されます。



この画面を表示するには：プラクティス > 静的解析 > 静的解析 - 違反 > [違反の数] (表の中) > [ファイル]

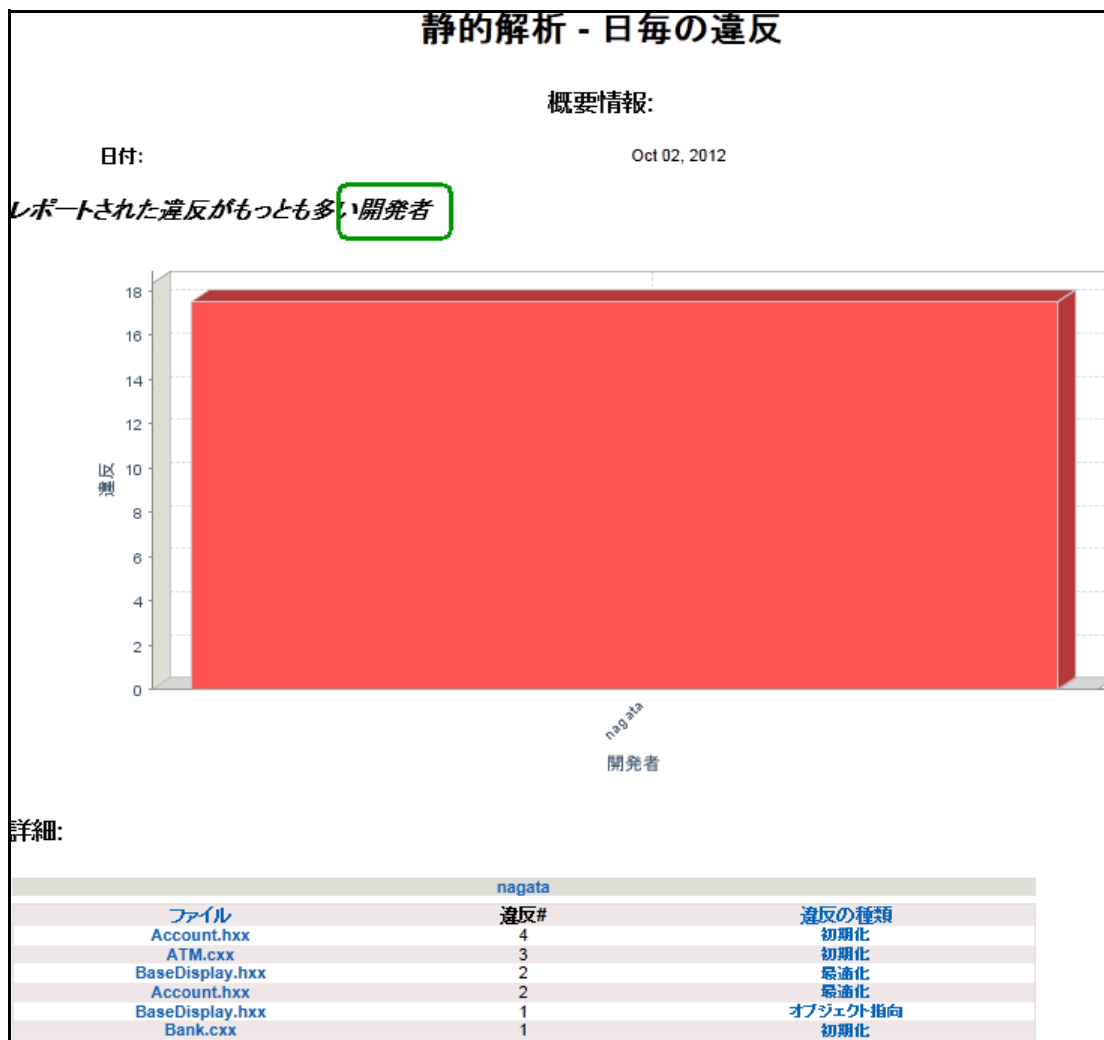
表のリンクをクリックするとビューが変化し、さらに詳しい情報を参照できます。

- [違反の種類] 列のヘッダー：違反タイプ別ビューに切り替わります。
- [開発者] 列のヘッダー：開発者別ビューに切り替わります。
- 違反タイプ名：[静的解析の違反 - 違反タイプ別] ページが表示されます。詳細については 44 ページ を参照してください。

- ファイル名：[静的解析の違反 - ファイル別] ページが表示されます。詳細については 45 ページ を参照してください。
- 開発者名：[静的解析の違反 - 開発者別] ページが表示されます。詳細については 45 ページ を参照してください。

静的解析の違反 - 開発者別

[静的解析の違反 - 開発者別] グラフの棒は、レポートされた違反数が多い開発者を表しています。上位 10 人までの開発者が表示されます。



この画面を表示するには：プラクティス > 静的解析 > 静的解析 - 違反 > [違反の数] (表の中) > [開発者] (表のヘッダー)

表のリンクをクリックするとビューが変化し、さらに詳しい情報を参照できます。

- [違反の種類] 列のヘッダー：違反タイプ別ビューに切り替わります。
- [ファイル] 列のヘッダー：ファイル別ビューに切り替わります。
- 違反タイプ名：[静的解析の違反 - 違反タイプ別] ページが表示されます。詳細については 44 ページ を参照してください。

- ファイル名：[静的解析の違反 - ファイル別] ページが表示されます。詳細については 45 ページ を参照してください。
- 開発者名：[静的解析の違反 - 開発者別] ページが表示されます。詳細については 45 ページ を参照してください。

[静的解析 - 違反の詳細]

[静的解析 - 違反の詳細] には 3 種類の表示があります。[静的解析の違反 - 違反タイプ別] ページでクリックしたリンクによって、以下のいずれかが表示されます。

- 違反タイプ：選択した日付と違反タイプについて、使用されたルールの数、違反されたルールの数、および違反の総数が表示されます。
- ファイル：選択した日付とファイルについて、使用されたルールの数、違反されたルールの数、および違反の総数が表示されます。
- 開発者：選択した日付と開発者について、使用されたルールの数、違反されたルールの数、および違反の総数が表示されます。

違反タイプ

[静的解析の違反 - 違反タイプ別] ページで違反タイプの名前をクリックすると、その違反タイプの [静的解析 - 違反の詳細] が表示されます。

図 2: 静的解析 - 違反の詳細 (違反の種類)

静的解析 - 違反の詳細				
概要情報:				
違反の種類:		初期化		
日付:		Oct 02, 2012		
使用されたルール総数:		90 (すべての違反タイプについて)		
違反されたルール:		2		
違反総数:		8		
8 発見された行				
ルール ID	行	ファイル	開発者	メッセージ
INIT-14	13	Account.hxx	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	14	Account.hxx	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	18	Account.hxx	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	19	Account.hxx	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-06	4	ATM.cxx	nagata	コンストラクタがすべてのメンバ変数を初期化していない: myCurrentAccount
INIT-14	6	ATM.cxx	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	7	ATM.cxx	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	6	Bank.cxx	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する

この画面を表示するには：プラクティス > 静的解析 > 静的解析 - 違反 > [違反の数] (表の中) > [違反タイプ名] (表のセクション タイトル、たとえば：保守性、グローバル静的解析、重複コードの検出など)

ファイル

[静的解析の違反 - 違反タイプ別] ページでファイルの名前をクリックすると、そのファイルの [静的解析 - 違反の詳細] が表示されます。

図 3: 静的解析 - 違反の詳細 (ファイル)

静的解析 - 違反の詳細					
概要情報:					
ファイル:	Account.hxx				
パス:	C:\Users\nagata\parasoft\workspace\ATM1001\Demo_dev3_nagata\ATM1002\include				
日付:	Oct 02, 2012				
使用されたルール総数:	90				
違反されたルール:	3				
違反総数:	7				
7 発見された行					
ルール ID	ルールカテゴリ	行	開発者	メッセージ	
INIT-14	初期化	19	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する	
INIT-14	初期化	18	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する	
OPT-17	最適化	16	nagata	クラス Account はインライン コンストラクタを定義している	
CODSTA-CPP-04	C++ コーディング規約	16	nagata	変換を可能にするコンストラクタ Account を explicit にする	
INIT-14	初期化	14	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する	
INIT-14	初期化	13	nagata	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する	
OPT-17	最適化	11	nagata	クラス Account はインライン コンストラクタを定義している	

この画面を表示するには: プラクティス > 静的解析 > 静的解析 - 違反 > [違反の数] (表の中) > [ファイル名]

開発者

[静的解析の違反 - 違反タイプ別] ページで開発者の名前をクリックすると、その開発者の [静的解析 - 違反の詳細] が表示されます (図 4)。

図 4: 静的解析 - 違反の詳細 (開発者)

静的解析 - 違反の詳細				
概要情報:				
開発者:		nagata		
日付:	Oct 02, 2012			
使用されたルール総数:	90			
違反されたルール:	8			
違反総数:	18			
18 発見された行				
ルール ID	ルールカテゴリ	行	ファイルへ	メッセージ
OPT-17	最適化	11	Account.hxx	クラス Account はインライン コンストラクタを定義している
OPT-17	最適化	16	Account.hxx	クラス Account はインライン コンストラクタを定義している
INIT-14	初期化	13	Account.hxx	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	初期化	14	Account.hxx	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	初期化	18	Account.hxx	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	初期化	19	Account.hxx	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
CODSTA-CPP-04	C++ コーディング規約	16	Account.hxx	変換を可能にするコンストラクタ Account を explicit にする
OPT-14	最適化	10	ATM.cxx	The parameter 'password' of function 'viewAccount' is passed by value
INIT-06	初期化	4	ATM.cxx	コンストラクタがすべてのメンバ変数を初期化していない: myCurrentAccount
INIT-14	初期化	6	ATM.cxx	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する
INIT-14	初期化	7	ATM.cxx	コンストラクタで代入している。代わりにコンストラクタ初期化リストを使用する

この画面を表示するには: プラクティス > 静的解析 > 静的解析 - 違反 > [違反の数] (表の中) > [開発者名]

[静的解析 - 違反の詳細] レポートには以下の情報が表示されます。括弧内はこのレポートの種類を表します。

概要情報セクション:

- **違反タイプ:** (違反タイプ) 違反されたルールの種類。
- **ファイル:** (ファイル) 違反がレポートされたファイルの名前。ファイル名のリンクをクリックすると、そのファイルのソースコードが表示されます。
- **パス:** (ファイル) 違反がレポートされたファイルのパス。
- **開発者:** (開発者) 違反に責任がある開発者の名前
- **日付:** テストが実行されて違反が発見された日。
- **使用されたルール総数:** 実行されたテストで使用されたルールの総数。このリンクをクリックすると、別のページが開いてすべてのルール ID が表示されます。47 ページの「使用された静的解析ルール」を参照してください。
- **違反されたルール:** 違反があったルールの数。このリンクをクリックすると、別のページが開き、違反があったルールのルール ID が表示されます。
- **違反総数:** 静的解析テストで検出された違反の総数。
- **ファイル履歴:** (ファイル) このリンクをクリックすると、ファイルの変更履歴 n ページが表示されます。
- **メッセージ:** 各違反のエラーメッセージ。

表のセクション

- **ルール ID:** ルールの ID。クリックすると、別のページが開いてそのルールの説明が表示されます。
- **行:** (違反タイプ および ファイル) 違反が検出されたコード行。クリックすると、ファイルのソース コードが表示され、違反があったコード行がハイライト表示されます。
- **ファイル:** (違反タイプ および 開発者) 違反があったファイルの名前。クリックすると、そのファイルのソース コードが表示されます。
- **開発者:** (違反タイプ および ファイル) エラーに責任がある開発者の名前。名前をクリックすると、その開発者についての統計情報が表示されます。
- **ルール カテゴリ:** (違反タイプ および ファイル) クリックすると、そのルール カテゴリの違反の統計情報が表示されます。

表の中でデータが表示される順序を変更するには、表のヘッダーをクリックします。

注意: 特定のファイルのソース コードおよびファイル履歴は、SourceScanner を実行した後に参照できるようになります。

使用された静的解析ルール

[使用された静的解析ルール] ページ (図 5) には、特定の日にテストされたすべてのルールのリストが表示されます。使用されたルールの総数はレポートのヘッダーに表示されます。

図 5: 使用された静的解析ルール

Static Analysis - Rules Used	
Summary Information:	
Violation Type: Property Files	
Date:	Aug 18, 2007
Total Rules Used:	698 (for all violation types)
Rules Violated:	11
Total Violations:	3604
698 Row(s) Found	
Rule ID ▲	Violations Triggered
AEP.ETP	0
AEP.RHTC	0

この画面を表示するには: プラクティス > 静的解析 > 静的解析 - 違反の詳細 > [使用されたルール総数]

ルール ID をクリックすると、別のページが開き、そのルールの説明、セキュリティとの関連、パラメーターなどの情報が表示されます。

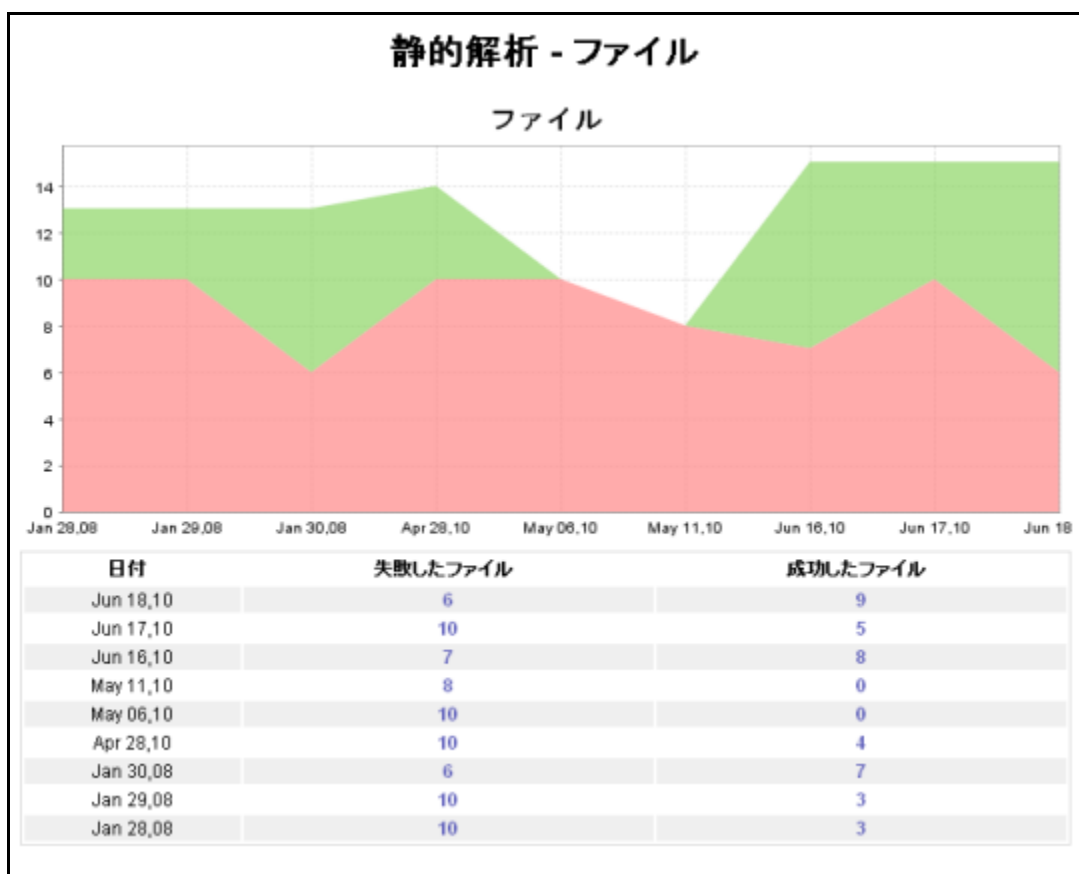
[静的解析 - 抑制] ページ

[静的解析 - 抑制] ページには、選択されたプロジェクトで抑制されているルールの数や、テスト中に違反が発生した、抑制されたルール違反の数などの情報が表示されます。詳細については 315 ページの「静的解析の抑制」を参照してください。

[静的解析 - ファイル] レポート

[静的解析 - ファイル] レポート (図 6) には、特定の期間に静的解析テストで失敗したテストの数と成功したテストの数が表示されます。プロジェクトが進行するにつれて、プロジェクトが静的解析に準拠しているかどうかの傾向が [静的解析 - ファイル] レポートによって明らかになります。

図 6: [静的解析 - ファイル] レポート



この画面を表示するには : プラクティス > 静的解析 > ファイル

グラフの下の [静的解析 - ファイル] の表では、指定の期間内のドロップ日ごとに、少なくとも 1 つの違反があるファイル (失敗したファイル) の数と違反がまったく検出されなかったファイル (成功したファイル) の数が表示されます。

次の列の下にあるリンクをクリックすると、さらに詳しい統計が表示されます。

- **失敗したファイル** [静的解析 - 日毎の違反] ページが開きます。詳細については 42 ページの「静的解析の違反 - ファイル別」を参照してください。
- **成功したファイル** [静的解析 - 違反のないファイル] ページが開きます。詳細については 49 ページの「[静的解析 - 違反のないファイル] レポート」を参照してください。

[静的解析 - 違反のないファイル] レポート

[静的解析 - 違反のないファイル] レポート (図 7) には、静的解析テストに成功したすべてのファイルが表示されます。このレポートはユーザーの次の疑問に回答します。

- テストに関係するルールは何か？
- 違反ゼロでテストをパスしたファイルはどれか？
- 誰がいつファイルを作成したのか？
- 誰がいつファイルを変更したのか？

図 7: [静的解析 - 違反のないファイル] レポート

静的解析 - 違反のないファイル		
概要情報:		
日付:	Jun 18, 2010	
ファイル:	9	
使用されたルール総数:	57	
		9 発見された行
[未テストのファイル]		
ファイル ▲	作成	最終変更日
OutputcliCmdCheck.cpp	2010-06-16	2010-06-18
OutputcliCommentString.h	2010-06-16	2010-06-18
OutputcliKeywordString.h	2010-06-16	2010-06-18
OutputcliOutputLine.h	2010-06-16	2010-06-18
OutputcliOutputString.h	2010-06-16	2010-06-18
OutputcliSrcCodeString.cpp	2010-06-16	2010-06-18
OutputcliSrcCodeString.h	2010-06-16	2010-06-18
OutputcliStdafx.cpp	2010-06-16	2010-06-16
OutputcliStdafx.h	2010-06-16	2010-06-17

この画面を表示するには: プラクティス > 静的解析 > 静的解析 - ファイル > [成功したファイル] (表)

[静的解析 - 違反のないファイル] ページは 2 つの部分から構成されます。

概要情報

- **日付** テストが実行された日。成功したすべてのファイルが表示されます。
- **ファイル** 指定の日付にテストをパスしたファイルの名前。
- **使用されたルール総数** 実行されたテストで使用されたルールの総数。クリックすると、[静的解析 - 使用されたルール] レポートが開きます。すべてのルールの ID が表示されます。

- **未テストのファイル** クリックすると、[静的解析 - テストされていないファイル] ページが開きます。表示中の日付にテストされなかったすべてのファイルが表示されます。

詳細情報

- **ファイル** 表示中の日付に実行されたテストで成功した各ファイルについて、パスとファイル名が表示されます。クリックすると、[静的解析 - 違反の詳細] ページが開きます。
- **作成** ファイルが作成された日。
- **最終変更日** ファイルが最後に変更された日。

表のデータをソートするには、表の列ヘッダーをクリックします。

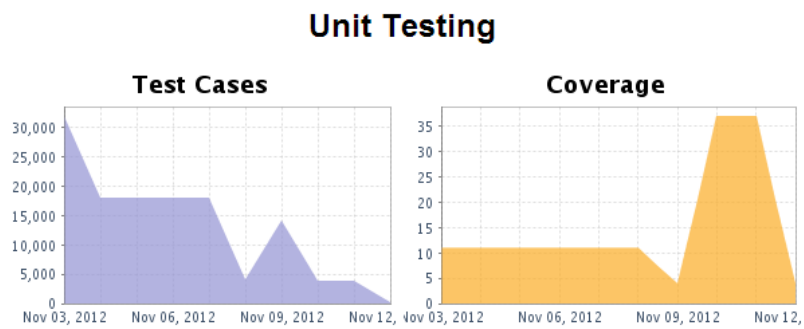
注意！ 特定のファイルのソース コードおよびファイル履歴は、SourceScanner を実行した後に参照できるようになります。

[単体テスト]レポート

[単体テスト]レポートは単体テスト結果の統計を組み合わせることで次の3種類のレポートとして表示します。

- [単体テスト]レポート
- [単体テスト テスト ケース]レポート
- [単体テスト カバレッジ]レポート

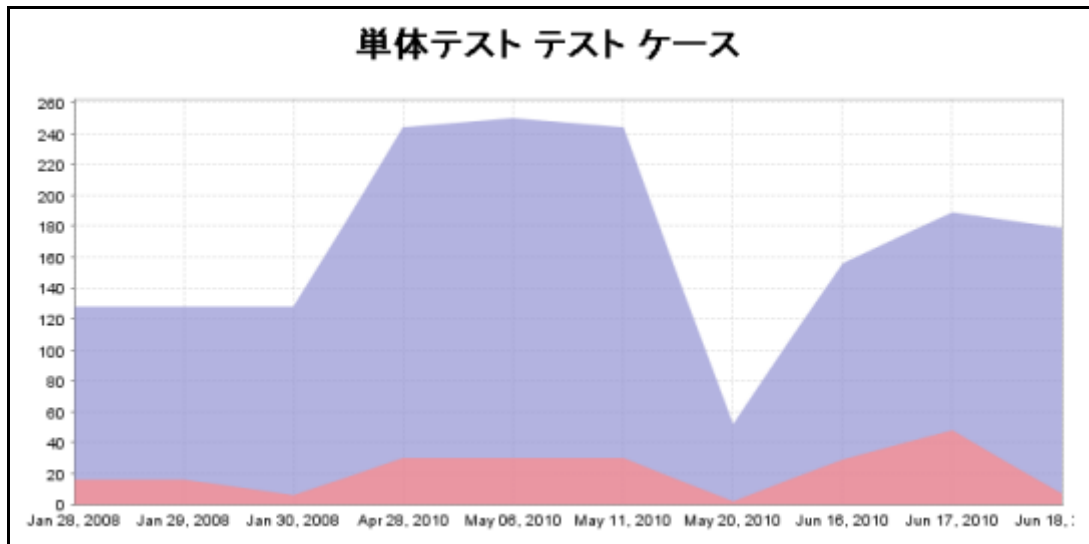
図 8: [単体テスト]レポート



この画面を表示するには: プラクティス > 単体テスト

[単体テスト テスト ケース] レポート

[単体テスト テスト ケース] レポートのグラフには、すべての単体テストについて、テスト ケースの総数と失敗したテスト ケースの総数が示されます。時間の経過につれて、実行されるテスト ケースの数は増加し、失敗は減少するべきです。テスト ケースの失敗は 0 であるべきです。



この画面を表示するには：[プラクティス](#) > [単体テスト](#) > [テスト ケース](#)

[単体テスト テスト ケース] レポートの下の表は、ドロップ日ごとのテスト ケースの総数と失敗したテスト ケースの数を示します。

日付	エラー	合計
Jun 18, 2010	7	179
Jun 17, 2010	48	189
Jun 16, 2010	29	156
May 20, 2010	2	52
May 11, 2010	30	244
May 06, 2010	30	250
Apr 28, 2010	30	244
Jan 30, 2008	6	128
Jan 29, 2008	16	128
Jan 28, 2008	16	128

[合計] 列の値をクリックすると、「[単体テスト - 詳細レポート - すべてのファイル] ページ」が表示されます。

[エラー] 列の値をクリックすると、「[単体テスト - 詳細レポート - 失敗したファイル] レポート」が表示されます。

[単体テスト - 詳細レポート - すべてのファイル] ページ

[単体テスト - 詳細レポート - すべてのファイル] ページでは、テスト ケースの実行、失敗、推定されるカバレッジ、テストされたメソッド、およびエラーがあるメソッドの数がファイルごとにレポートされます。

単体テスト - 詳細レポート - すべてのファイル

概要情報:

日付: Jun 18, 2010

失敗があるファイル: 5
 失敗がないファイル: 6
 テスト ケース実行: 179
 テスト ケース実行が失敗 (エラー): 7
 相違テスト ケース: 51
 失敗した相違テスト ケース: 7

エラー サマリー

Failure 7

[未テストのファイル]

ソート基準:
 ファイル ▲
 テストケース実行 失敗 推定されるカバレッジ テストされたメソッド エラーがあるメソッド

ファイル: CmdCheck.cpp テスト ケース実行: 43 失敗: 0 推定されるカバレッジ: 85% テストされたメソッド: 0 エラーがあるメソッド: 0
ファイル: CommentString.cpp テスト ケース実行: 32 失敗: 0 推定されるカバレッジ: 100% テストされたメソッド: 0 エラーがあるメソッド: 0
ファイル: KeywordString.cpp テスト ケース実行: 24 失敗: 0 推定されるカバレッジ: 100% テストされたメソッド: 0 エラーがあるメソッド: 0
ファイル: OutputCil.cpp テスト ケース実行: 6 失敗: 0 推定されるカバレッジ: 100% テストされたメソッド: 0 エラーがあるメソッド: 0
ファイル: OutputLine.cpp テスト ケース実行: 18 失敗: 0 推定されるカバレッジ: 100% テストされたメソッド: 0 エラーがあるメソッド: 0

この画面を表示するには：プラクティス > 単体テスト > テスト ケース > [合計] (グラフまたは表)

ファイルをクリックすると、「[単体テスト カバレッジ] レポート」が表示されます。

[単体テスト - ファイルの詳細] レポート

このレポートには、特定のファイルの詳細が表示されます。

Unit Tests - File Detailed Report

Summary Information:

File:	CppExecutionResult.java	[File History]
Path:	common/com.parasoft.xtest.execution.api.cpp/src/com/parasoft/xtest/execution/api/cpp/results	
Date:	Nov 12, 2012	
Test cases run:	4	
Test cases run failed (Errors):	0	
Coverage:	73% [29/40]	
Methods tested:	4	
Methods not tested:	0	

Methods not tested [?](#)

Not available. Project metrics must be computed to obtain this information.

Methods tested

Method ^	Test Cases	Errors found	Coverage
getProblemReportIdentifiers()	1	0	75%
getReqIdentifiers()	1	0	75%
getTaskIdentifiers()	1	0	75%
N/A	1	0	N/A

この画面を表示するには：プラクティス > 単体テスト > テスト ケース > [合計] > [単体テストファイル名]

ファイル名をクリックすると、ソース コードが表示されます。

[メソッド] 列の値をクリックすると、テスト ケースの詳細が開きます。

[単体テスト - 詳細レポート - 失敗したファイル] レポート

このレポートは、単体テストが失敗したファイルの概要を表示します。

単体テスト - 詳細レポート - 失敗したファイル

概要情報:

日付: Jun 18, 2010

失敗があるファイル: 5

テストケース実行: 179

テストケース実行が失敗 (エラー): 7

相違テストケース: 51

失敗した相違テストケース: 7

エラー サマリー

Failure 7

[未テストのファイル]

ソート基準:

ファイル ▲

テストケース実行 失敗 推定されるカバレッジ テストされたメソッド エラーがあるメソッド

ファイル: [SrcCodeString.cpp](#)

テストケース実行: 34 失敗: 1 推定されるカバレッジ: 100% テストされたメソッド: 0 エラーがあるメソッド: 0

行	エラー	開発者	オカレンスの数
152	Failure	不明	1

ファイル: [TestSuite_Output.cpp](#)

テストケース実行: 3 失敗: 3 推定されるカバレッジ: 85% テストされたメソッド: 0 エラーがあるメソッド: 0

行	エラー	開発者	オカレンスの数
44	Failure	不明	1
58	Failure	不明	1
72	Failure	不明	1

ファイル: [TestSuite_OutputArea.cpp](#)

テストケース実行: 1 失敗: 1 推定されるカバレッジ: 85% テストされたメソッド: 0 エラーがあるメソッド: 0

行	エラー	開発者	オカレンスの数
49	Failure	不明	1

この画面を表示するには: プラクティス > 単体テスト > テスト ケース > [エラーの数] (グラフ)

ファイル名をクリックすると、[単体テスト カバレッジ] レポートが開きます。

[行] 列の値をクリックすると、ソース コードが表示されます。

[単体テスト - 詳細レポート - 失敗したファイル] ページ

このレポートには、単体テストが失敗した特定のファイルに関する詳細な情報が表示されます。表示されるファイルの情報は、[単体テスト - ファイルの詳細] レポートに表示されるものと同じです。

Unit Tests - File Detailed Report

Summary Information:

File:	AjaxTestingTool.java	[File History]
Path:	webtool/ajax	
Date:	Sep 17, 2012	
Test cases run:	28	
Test cases run failed (Errors):	10	
Coverage:	46% [100/218]	
Methods tested:	1	
Methods not tested:	0	

Test Cases Failed			
Line ^	Error	Developer(s)	Method
739	Failure	tsaucedo	N/A
1062	Failure	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A
1216	webtool.browsercontroller.BrowserException	tsaucedo	N/A

Methods not tested ?			
Not available. Project metrics must be computed to obtain this information.			

Methods tested			
Method ^	Test Cases	Errors found	
N/A	28	10	

この画面を表示するには：プラクティス > 単体テスト > テスト ケース > [エラーの数] (グラフ)

概要情報

- 日付 テストが実行された日。
- 失敗があるファイル 失敗があったファイルの数。
- 失敗がないファイル 失敗がなかったファイルの数。
- テスト ケース実行 実行されたテスト ケースの数。
- テスト ケース実行が失敗 (エラー) 失敗したテスト ケースの数。
- 相違テスト ケース 相違があるテスト ケースの数。
- 失敗した相違テスト ケース 失敗した、相違があるテスト ケースの数。

エラー サマリー

リンクをクリックして、さらに詳しい情報にアクセスすることができます。


図 9: [単体テスト - 詳細レポート - 失敗したファイル] レポート

単体テスト - 詳細レポート - 失敗したファイル

概要情報:


日付: Jun 18, 2010


失敗があるファイル: 5
 テスト ケース実行: 179
 テスト ケース実行が失敗 (エラー): 7
 相違テスト ケース: 51
 失敗した相違テスト ケース: 7

エラー サマリー 


Failure 7

[未テストのファイル]


ソート基準:
 ファイル ▲
 テスト ケース実行 失敗 推定されるカバレッジ  テストされたメソッド エラーがあるメソッド

ファイル: SrcCodeString.cpp 
 テスト ケース実行: 34 失敗: 1 推定されるカバレッジ: 100% テストされたメソッド: 0 エラーがあるメソッド: 0

行	エラー	開発者	オカレンスの数
152	Failure	不明	1

ファイル: TestSuite_Output.cpp 
 テスト ケース実行: 3 失敗: 3 推定されるカバレッジ: 85% テストされたメソッド: 0 エラーがあるメソッド: 0

行	エラー	開発者	オカレンスの数
44	Failure	不明	1
58	Failure	不明	1
72	Failure	不明	1

ファイル: TestSuite_OutputArea.cpp 
 テスト ケース実行: 1 失敗: 1 推定されるカバレッジ: 85% テストされたメソッド: 0 エラーがあるメソッド: 0

行	エラー	開発者	オカレンスの数
49	Failure	不明	1

この画面を表示するには:

方法 1: プラクティス > 単体テスト > テスト ケース > [失敗したファイル] > [テスト数合計] (グラフまたは表) > [ファイル名] (ページ下部に表示された表から)

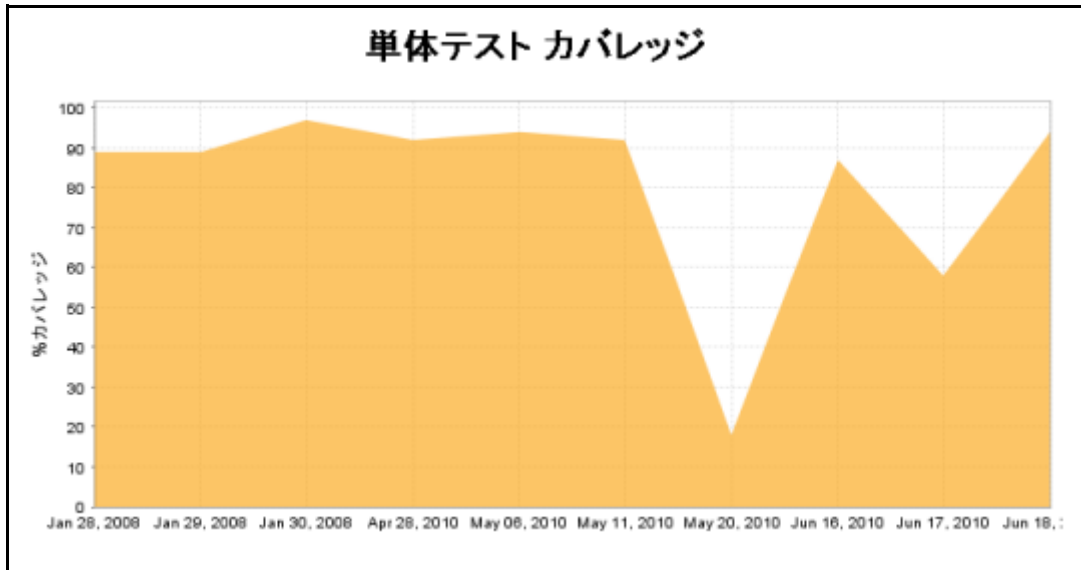
方法 2: プラクティス > 単体テスト > テスト ケース > [失敗したファイル] > [エラー数] (表) > [ファイル名] (ページ下部に表示された表から)

[単体テスト カバレッジ] レポート

[単体テスト カバレッジ] レポートは、時間の経過に伴う単体テスト カバレッジの進捗をモニターするのに役立ちます。カバレッジのパーセンテージは時間の経過とともに上昇するべきです。カバレッジ率が上昇していない場合、それはテストが適切に作成されていないことを意味します。[単体テスト カバレッジ] レポートのグラフは、次の情報を表示します。

- 実際にテストされたコードの量
- 選択されたプロジェクト ファイルの単体テスト カバレッジ
- テスト済みのユニット、およびまだテストが必要なユニットの数

グラフには行カバレッジだけが表示されます。MC/DC やブランチ カバレッジなどは表示されません。カバレッジは、テスト対象として選択されたソース コード行数の合計に対する、テストが成功した行数のパーセンテージです。



この画面を表示するには：プラクティス > 単体テスト > カバレッジ

[単体テスト カバレッジ] レポートの下の表はドロップ日ごとのカバレッジ率を示します。テストされたユニットの総数とまだテストされていないユニットの数も表示されます。

日付	%カバレッジ	テストされたユニット	テスト対象ユニット
Jun 18, 2010	94	303	323
Jun 17, 2010	58	187	325
Jun 16, 2010	87	26	30
May 20, 2010	18	9	51
May 11, 2010	92	94	102
May 06, 2010	94	116	124
Apr 28, 2010	92	94	102
Jan 30, 2008	97	60	62
Jan 29, 2008	89	55	62
Jan 28, 2008	89	55	62

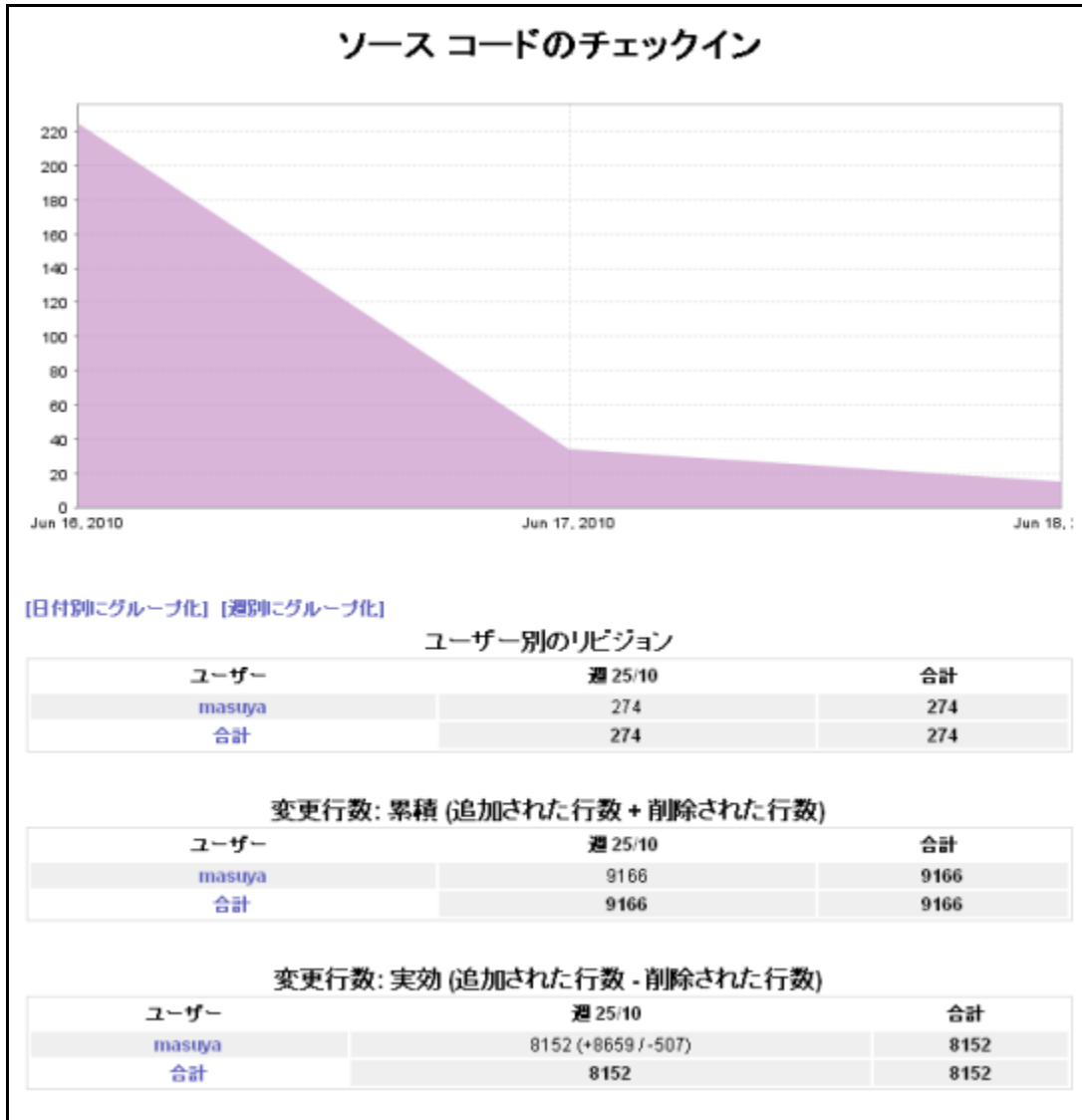
[単体テスト カバレッジ] レポートでグラフの中または表の日付をクリックすると、[単体テスト - 詳細レポート - すべてのファイル] ページが開き、単体テストに含まれるファイルについての詳細情報を参照できます。

[単体テスト - 詳細レポート - すべてのファイル]

[単体テスト - 詳細レポート - すべてのファイル] ページに表示されている任意のテスト ケースをクリックすると、単体テストに含まれるファイルについての詳細情報を参照することができます。このページに表示されるデータの詳細については「[単体テスト カバレッジ] レポート」を参照してください。

[ソースコードのチェックイン]レポート

レポートビューで [プラクティス] > [ソースコードのチェックイン] をクリックすると、[ソースコードのチェックイン] レポートが開きます。このレポートには、指定のプロジェクトに関連するソース管理ファイルの改訂回数が、指定の日付範囲の日ごとに表示されます。



- [ソースコードのチェックイン] グラフの特定の日付のエリアをクリックすると、その日に更新されたすべてのファイルのソース管理システムの概要レポートが開きます。
- [日付別にグループ化] リンクをクリックすると、各ユーザーの改訂が日付順でソートされて表示されます。
- [週別にグループ化] リンクをクリックすると、各ユーザーの改訂がグラフに表示された日付を基に週ごとにグループ化されて表示されます。
- ユーザー名をクリックすると、そのユーザーによって更新されたすべてのファイルのソース管理システムの概要レポートが表示されます。

ソース管理システムの概要

[ソース管理システムの概要] レポートは、選択された日付にソース管理システムで変更されたファイルおよびディレクトリを表示します。

Source Control Summary

Date: Nov 22, 2013
User: Any Valuenull
File Revisions: 9
Lines Changed: 1713 (+856/-857)
Token Changed: 0 (+0/-0)

[Files](#) [Directories](#)

Files Updates

File	11/22/13	Total
com.parasoft.grs.rserver/pom.xml	1	1
com.parasoft.grs.rserver/src/main/java/com/parasoft/grs/rserver/console/DatabaseCreatorConsole.java	1	1
com.parasoft.sdm.concerto/build.xml	5	5
com.parasoft.sdm.concerto/old_build/install_header.sh	1	1
com.parasoft.sdm.shared.lib/pom.xml	1	1
Total	9	9

Lines changed

File	11/22/13	Total
com.parasoft.grs.rserver/pom.xml	+5/-0	+5/-0
com.parasoft.grs.rserver/src/main/java/com/parasoft/grs/rserver/console/DatabaseCreatorConsole.java	+33/-34	+33/-34
com.parasoft.sdm.concerto/build.xml	+26/-21	+26/-21
com.parasoft.sdm.concerto/old_build/install_header.sh	+0/-5	+0/-5
com.parasoft.sdm.shared.lib/pom.xml	+792/-797	+792/-797
Total	+856/-857	+856/-857

Tokens Changed

File	11/22/13	Total
com.parasoft.grs.rserver/pom.xml	+0/-0	+0/-0
com.parasoft.grs.rserver/src/main/java/com/parasoft/grs/rserver/console/DatabaseCreatorConsole.java	+0/-0	+0/-0
com.parasoft.sdm.concerto/build.xml	+0/-0	+0/-0
com.parasoft.sdm.concerto/old_build/install_header.sh	+0/-0	+0/-0
com.parasoft.sdm.shared.lib/pom.xml	+0/-0	+0/-0
Total	+0/-0	+0/-0

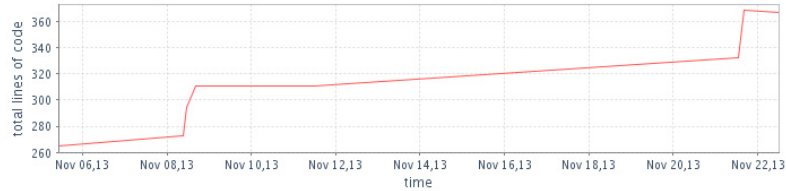
- [ファイル] または [ディレクトリ] をクリックすると、ファイル ビューまたはディレクトリ ビューに切り替わります。

- ファイルビューで [ファイル] 列のリンクをクリックすると詳細ビューが開き、各改訂で追加されたコード行数が表示されます。

File Details

File: [com.parasoft.grs.rserver/src/main/java/com/parasoft/grs/rserver/console/DatabaseCreatorConsole.java](#)

DatabaseCreatorConsole.java



Date	Version	User	Lines changed	Tokens Changed	Comment
11/22/13 11:53 AM	62de0887de89f069b1819b7807bb87e6c933c0b4	Ted Kim	+33/-34	+0/-0	@task 57024 - Adding setting install flags after database is created/updated
11/21/13 4:20 PM	1528495e67c2cb94ce790431c7c3343f92c24361	Ted Kim	+54/-18	+0/-0	@task 57024 - Constructed bash script for creating/updating database settings during first install
11/21/13 1:22 PM	a771dea0911d5fe4e881a4f31008e313d74301f2	Ted Kim	+21/-0	+0/-0	@task 57023 - Setting up new dtp-fresh-install flag in PSTRootConfig
11/11/13 11:52 AM	f096c0181b7850be864c0e0d90b25337e75dd9c	Ted Kim	+26/-26	+0/-0	Resolving static analysis violations
11/8/13 4:03 PM	2fddb6c8ed60e98ed0c984f00529768959e20a3f	Ted Kim	+18/-1	+0/-0	Saving database configuration after database is created
11/8/13 11:28 AM	4016f51f987d6bb9683e55dbf98419c43b82b83d	Ted Kim	+34/-13	+0/-0	Fixing console read inconsistencies
11/8/13 9:13 AM	dea628dc0df4ed2a91c6c49f71ee41880e66bd8c	Ted Kim	+8/-0	+0/-0	Resolving build failures
11/5/13 10:13 AM	ef6ac4ad3773023e86346dae8ee7b61194e87cd	Ted Kim	+265/-0	+0/-0	@task 57023 - Refactored database creation code to allow creation from terminal

- ディレクトリビューで [ディレクトリ] 列のリンクをクリックすると、選択されたディレクトリのソース管理システムの概要レポートが開きます。

ビルドレポート

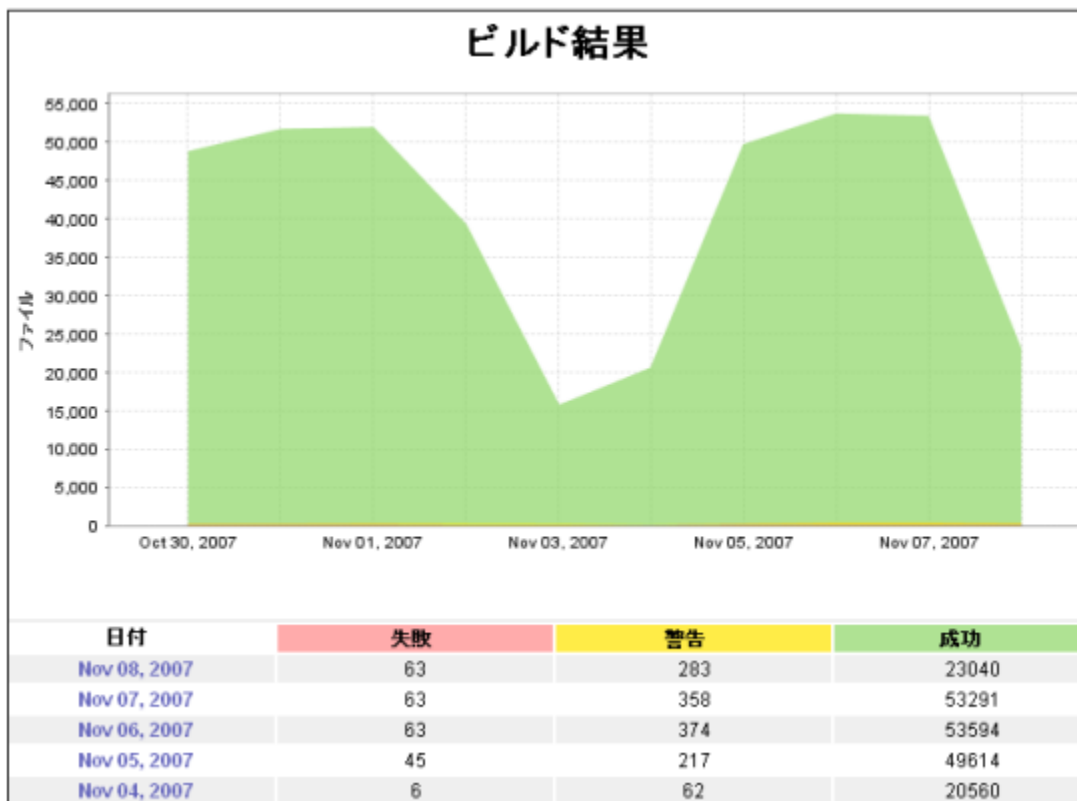
ビルド モニターは、コンパイルとビルドのプロセスを追跡し、コンパイル出力を解析し、結果を Report Center に送ります。このデータはオン デマンドで Report Center によってレポートされ、ユーザーはビルド プロセスの最終結果を検証し、エラーが発生した場所を正確に特定することができます。

[ビルド結果] レポート

[ビルド結果] レポート (7 ページの「ビルド結果ウィジェット」を参照) には、リストされたドロップ日ごとに、次のビルド情報が表示されます。

- 失敗したファイルの数
- 警告があり、未完了であるファイルの数
- 成功したファイルとモジュールの数

オプション バーから選択されたプロジェクトと周期に基づいてデータが表示されます。[ビルド結果] レポート



グラフまたは表で適切な日付をクリックすると、[プラットフォーム別ビルド] レポートが表示されます。

[プラットフォーム別ビルド] レポート

[プラットフォーム別ビルド] レポートには、選択された日付にビルドが実行されたプラットフォームがリストされます。リストされたプラットフォームごとに、アーキテクチャとバージョン番号が表示されるほか、ビルド プロセス中に失敗したファイル、警告があったファイル、成功したファイルの数も表示されます。

OS Name ▲	OS Arch	OS Version	Fail	Warning	Pass	Total
Linux	i386	2.4.21-47.0.1.ELsmp	0	10	9856	9866
Linux	amd64	2.6.22.9-61.fc6	1	0	2378	2379
Linux	i686	2.2.24-6.2.3 #1 Fri Mar 14 09:00:20 EST 2003	0	7	8	15
Windows XP	intel 6 0xa00	Version 5.1 (Build 2600: Dodatek Service Pack 2)	0	6	801	807
Windows XP	x86	5.1	44	188	35770	36002
Windows XP	intel 6 0x801	Version 5.1 (Build 2600: Dodatek Service Pack 2)	0	6	801	807
Total			45	217	49614	49876

[プラットフォーム別ビルド] レポートには次の情報が表示されます。

- **プロジェクト** ドロップダウン リストから選択された、ビルドが実行されたプロジェクトの名前。
- **日付** [ビルド結果] レポートから選択されたビルドの日付。
- **OS 名** ビルドが実行されたマシンにインストールされているオペレーティング システムの名前。クリックすると、[ビルド実行の詳細] レポートが表示されます。詳細については 64 ページの「[ビルド実行の詳細] ページ」を参照してください。
- **OS アーキテクチャ** ビルドが実行されたプロセッサの種類。
- **OS バージョン** ビルドが実行されたオペレーティング システムのバージョン。
- **失敗** 選択されたビルドについて、リストされたプラットフォームで失敗したファイルの数。クリックすると、[ビルド実行の詳細] レポートが表示されます。詳細については 64 ページの「[ビルド実行の詳細] ページ」を参照してください。
- **警告** 選択されたビルドについて、リストされたプラットフォームで警告があったファイルの数。クリックすると、[ビルド実行の詳細] レポートが表示されます。詳細については 64 ページの「[ビルド実行の詳細] ページ」を参照してください。
- **成功** 選択されたビルドについて、リストされたプラットフォームで成功したファイルの数。クリックすると、[ビルド実行の詳細] レポートが表示されます。詳細については 64 ページの「[ビルド実行の詳細] ページ」を参照してください。
- **合計 (列)** 選択されたビルドについて、リストされたプラットフォームでのビルドに含まれるファイルの数。
- **合計 (行)** すべてのプラットフォームにおける、失敗したビルド ファイル、警告があったビルド ファイル、成功したビルド ファイル、実行されたビルド ファイルの合計。クリックすると、[ビルド実行の詳細] レポートが表示されます。詳細については 64 ページの「[ビルド実行の詳細] ページ」を参照してください。

[プラットフォーム別ビルド] レポート の表で、[OS 名]、[失敗]、[警告]、[成功]、または [合計] リンクを介して適切なプラットフォームをクリックすると、[ビルド実行の詳細] レポートが表示されます。

[ビルド実行の詳細] ページ

[ビルド実行の詳細] ページ には、選択された日付に選択されたプラットフォームで実行されたビルドが表示されます。また、各ビルドに含まれるファイルも表示されます。ビルド中に失敗、警告、成功したファイルの数と、ファイルの総数を参照できます。

Build Execution Details							
Build Name ▲	Machine	OS Name/OS Arch	Tool	Fail	Warning	Pass	Total
.Test compilation	panama	Windows XP/x86	Build Logger	0	153	1637	1790
.Test compilation	panama	Windows XP/x86	Build Logger	0	90	1626	1716
Compiling of Axis2Integration	basilisk.parasoft.com	Windows XP/x86	Build Monitor	0	0	5	5
Compiling of BealIntegration	basilisk.parasoft.com	Windows XP/x86	Build Monitor	0	0	115	115
Compiling of bpeLCore	GECKO	Windows XP/x86	Build Monitor	0	2	296	298
Compiling of bpeLEngine	GECKO	Windows XP/x86	Build Monitor	0	2	349	351
Compiling of ComParasoft	terrapin.parasoft.com	Windows XP/x86	Build Monitor	12	0	1207	1219
Compiling of IONA-Integration	basilisk.parasoft.com	Windows XP/x86	Build Monitor	0	0	1	1
Compiling of Software AG Integration	basilisk.parasoft.com	Windows XP/x86	Build Monitor	0	0	8	8
Compiling of VMTTools	basilisk.parasoft.com	Windows XP/x86	Build Monitor	0	0	25	25
Compiling of WebKing	terrapin.parasoft.com	Windows XP/x86	Build Monitor	51	2	5097	5150
Compiling of WebSphereMQ	basilisk.parasoft.com	Windows XP/x86	Build Monitor	0	0	3	3
Compiling of WS-I Testing Tools	basilisk.parasoft.com	Windows XP/x86	Build Monitor	0	1	434	435
Compiling of WSS4J	basilisk.parasoft.com	Windows XP/x86	Build Monitor	0	4	415	419
Total				63	254	11218	11535

[ビルド実行の詳細] ページには次の情報が表示されます。

- **ビルド名** 選択された日付に選択されたプラットフォームで実行されたビルドの名前。クリックすると、[ビルド モジュールの詳細] ページが開き、すべてのビルド モジュールと各モジュール中のファイルがリストされます。詳細については 65 ページの「[ビルド モジュールの詳細] レポート」を参照してください。
- **マシン** ビルドが実行されたマシンの名前。
- **OS 名 / OS アーキテクチャ** ビルドが実行されたマシンにインストールされているオペレーティング システムの名前。
- **ツール** ビルドを実行するために使用されたツールの名前。
- **失敗** 選択されたプラットフォームのビルド中で失敗したファイルの数。クリックすると、[ビルド モジュールの詳細] ページが開き、選択された日付の選択されたビルドで失敗したファイルだけが表示されます。詳細については 65 ページの「[ビルド モジュールの詳細] レポート」を参照してください。
- **警告** 選択されたプラットフォームのビルド中で警告があったファイルの数。クリックすると、[ビルド モジュールの詳細] ページが開き、選択された日付の選択されたビルドで警告があったファイルだけが表示されます。詳細については 65 ページの「[ビルド モジュールの詳細] レポート」を参照してください。

- **成功** 選択されたプラットフォームのビルド中で成功したファイルの数。クリックすると、[ビルド モジュールの詳細] ページが開き、選択された日付の選択されたビルドで成功したファイルだけが表示されます。詳細については 65 ページの「[ビルド モジュールの詳細] レポート」を参照してください。
- **合計** 選択されたプラットフォームのビルドに含まれるファイルの総数。

[ビルド実行の詳細] レポートから詳細情報にアクセスする

[ビルド実行の詳細] の表で、[OS 名]、[失敗]、[警告]、[成功]、または [合計] リンクを介して適切な実行の詳細をクリックすると、[ビルド モジュールの詳細] レポートが表示されます。

[ビルド モジュールの詳細] レポート

[ビルド モジュールの詳細] レポートには、選択されたビルドに含まれる各モジュールがリストされます。

図 10: [ビルド モジュールの詳細] レポート

Build Modules Details		
Building wizard\$xml: 1 (0, 0, 1)		
File Name ▼	Messages #	Status
XMLRuleWriter.java	1	Pass
Building wizard@test: 2 (0, 0, 2)		
File Name ▼	Messages #	Status
TestHarness.java	1	Pass
DictionaryCombiner.java	1	Pass
Building webtool\ecmascript: 9 (1, 1, 7)		
File Name ▼	Messages #	Status
NodeReconstructor.java	1	Pass
ECMAScriptParserTokenManager.java	1	Pass
ECMAScriptParserTest.java	1	Pass
ECMAScriptParserConstants.java	1	Pass
ECMAScriptParser.java	3	Warning
+ C:\home\devtest\webtool_common\workspace\WebKing\webtool\ecmascript\ECMAScriptParser.java:3656: warning: as of...		
+ C:\home\devtest\webtool_common\workspace\WebKing\webtool\ecmascript\ECMAScriptParser.java:3656: warning: as of...		
+ C:\home\devtest\webtool_common\workspace\WebKing\webtool\ecmascript\ECMAScriptParser.java:3657: warning: as of...		
ECMAParserTest.java	1	Pass
ECMALinkFinder.java	1	Fail
+ C:\home\devtest\webtool_common\workspace\WebKing\webtool\ecmascript\ECMALinkFinder.java:100: package com.paras...		
ECMABeautifier.java	1	Pass
ECMA.java	1	Pass

この画面を表示するには: ビルド > ビルド結果 > [日付] (グラフまたは表) > [プラットフォーム別ビルド] レポートの表 [ビルド名] または [実行の統計] > [ビルド実行の詳細] レポートの表 [ビルド名] または [実行の統計]

図 10 の先頭のモジュールを検証して、利用可能な情報についての理解を深めましょう。

Building wizard\xml: 1 (0, 0, 1)

- **Building wizard** は、モジュールがあるディレクトリの名前です。
- **xml** はモジュール名です。
- **1** は、モジュール内にあるファイルの数を表します。
- **(0, 0, 1)** は、失敗したファイル、警告があったファイル、および成功したファイルの数をそれぞれ表します。

リストされたモジュールごとに、そのモジュールのために作成されたすべてのファイルが表にリストされます。ファイルごとに、生成されたメッセージの数とステータスも表示されます。

場合によっては、実際のメッセージが表中に表示されることもあります (図 10)。メッセージは次のように表示されます。

- **赤いフォント:** 失敗のファイル ステータスのみ。ビルド中に発見された問題を説明します。
- **黄色のフォント:** 警告のステータスのみ。ビルド中に小さな問題が検出されたことを表します。

[ビルド モジュールの詳細] レポート (図 10) には次の情報が表示されます。

- **ファイル名** ビルドに含まれるファイルの名前。[ビルド実行の詳細] レポートで選択されたリンクによって、[ビルド モジュールの詳細] レポートに表示されるファイルが決まります。次のビューがあります。
 - 特定の **ビルド名** をクリックすると、すべてのモジュールがリストされます。
 - **[失敗]** 列の数字をクリックすると、失敗のファイルを含むビルドのモジュールだけが表示されます。
 - **[警告]** 列の数字をクリックすると警告のファイルを含むビルドのモジュールだけが表示されます。
 - **[成功]** 列の数字をクリックすると、成功のファイルを含むビルドのモジュールだけが表示されます。

[ビルド モジュールの詳細] レポートでは (SourceScanner を適切に構成している場合)、ファイルの名前をクリックして [コンパイルされたファイルの詳細] (図 11) を表示し、ファイルのステータス、失敗または警告のある行番号、および対応するメッセージを参照することができます。

図 11: [コンパイルされたファイルの詳細]

Compiled Files Details			
File Name	Status	Line	Message
TestSuiteReference.java	Fail	378	C:\home\devtest\workspace\WebKing\workspace\test\TestSuiteReference.java

- **メッセージ #** ファイルに関連するメッセージの数がリストされます。失敗または警告ステータスのファイルに関連するメッセージの場合、ファイル名の下にメッセージが表示されます (図 12)。

図 12: [ビルド モジュールの詳細] - メッセージ #

Building webtooltest: 111 (5, 0, 106)		
File Name ▲	Messages #	Status
TestSuite.java	4	Fail
+ C:\home\devtest\webtool_common\workspace\WebKing\webtool\test\TestSuite.java:167: package com.parasoft.wspolic...		
+ C:\home\devtest\webtool_common\workspace\WebKing\webtool\test\TestSuite.java:2093: cannot find symbol		
+ C:\home\devtest\webtool_common\workspace\WebKing\webtool\test\TestSuite.java:2210: cannot find symbol		
+ C:\home\devtest\webtool_common\workspace\WebKing\webtool\test\TestSuite.java:2783: cannot find symbol		

- ステータス 失敗、警告、または成功。

コードレビュー レポート

Parasoft Test ソリューション (Parasoft Jtest、Parasoft C++test、Parasoft dotTEST、Parasoft SOAtest) のコードレビュー機能は、自動化されたコードレビューを容易にします。Report Center はこれらのツールのコードレビューデータを格納し、多彩なコードレビューレポートを提供します。レポートは、プロジェクトで行われるコードレビュープロセスを視覚化し、統計的データを提供します。

以下のタイプのコードレビューレポートがあります。

- コードレビュー アクティビティ：特定期間の統計値を表示します。
- コードレビュー ステータス：最新のステータスを表示します。
- アーキテクト ダッシュボードのコードレビュー サマリー グラフ

コードレビュー アクティビティ

コードレビュー アクティビティ レポートは、選択されたプロジェクトの指定された期間内におけるコードレビュープロセスの統計値を表示します。

Code Review Activity

Search Details: Review modified from: 2012-11-03; Review modified to: 2012-11-13; Saved Search: -- CUSTOMIZED --

By Author/Reviewer By Reviewer/Author By Requirement/Task

Author ^?	Reviewer?	New To Review?	To Review Done	Reviewer?				Time spent?	New To Fix?	To Fix Done	Author?			
				0-5	6-8	9-15	15+				0-5	6-8	9-15	15+
baranov	jakubiak	18	0	0	0	0	0	0	0	0	0	0	0	0
benken	bmerdian	175	42	17	1	24	0	0	0	0	0	0	0	0
	bmerdian	351	351	82	56	51	162	1h 51m 1s	0	0	0	0	0	0
>	cturek	10484	0	0	0	0	0		0	0	0	0	0	0
	daniel	10	0	0	0	0	0		0	0	0	0	0	0
	grigorb	23	0	0	0	0	0		0	0	0	0	0	0
>	humphrey	8	0	0	0	0	0		0	0	0	0	0	0
	jakubiak	186	120	48	45	27	0		0	0	0	0	0	0
	januszt	1	0	0	0	0	0		0	0	0	0	0	0
	iez	8	0	0	0	0	0		0	0	0	0	0	0
>	imadru	25	2	2	0	0	0	2m 31s	0	0	0	0	0	0
>	koczis	94	63	49	2	12	0	14m 29s	2	0	0	0	0	0
	kuba	5	0	0	0	0	0		2	2	0	0	0	2
	marcin	51	51	0	0	0	51		0	0	0	0	0	0
>	mestrada	61	0	0	0	0	0		0	0	0	0	0	0
>	mirage	4	1	0	1	0	0		2	1	0	1	0	0
>	mirek	6	0	0	0	0	0		0	0	0	0	0	0
>	mlove	411	0	0	0	0	0		0	0	0	0	0	0
	mmocko	29	29	1	0	2	26	18m 2s	0	0	0	0	0	0
	msiegel	38	0	0	0	0	0		0	0	0	0	0	0
	mstaron	6	0	0	0	0	0		0	0	0	0	0	0
>	mtran	13	0	0	0	0	0	2m 25s	1	0	0	0	0	0
	pawelf	173	0	0	0	0	0		0	0	0	0	0	0

デフォルト ビューの各行は、作成者 / レビューアーのペアごとにコードレビュー アクティビティの統計値を表示します。各行には、レビューアーに関する以下の統計値がグループ化されて表示されません。

- **レビュー待ち** 指定された期間内に特定のレビューアー向けに作成された新規コードレビューの数

- **レビュー済** 指定された期間内に承認済みまたは完了としてマークされたコード レビューの数
- **経過期間** レビューアーがレビューを完了するまでに経過した日数
- **消費時間** コード レビュー アクションに実際かけられた時間。Parasoft Test と統合された Eclipse または Visual Studio でソース コードを参照した時間です。

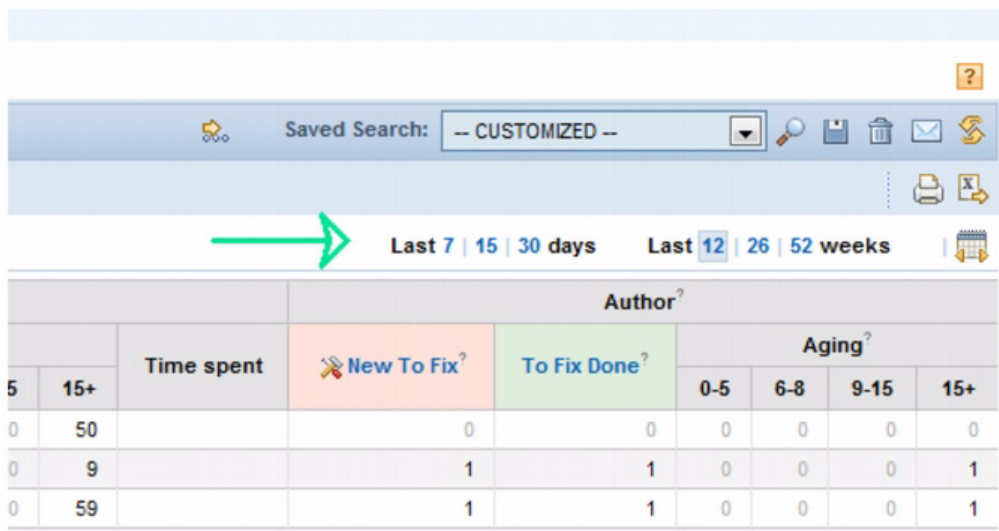
各行には作成者に関する以下の統計値がグループ化されて表示されます。

- **修正待ち** 指定された期間内に特定のレビューアー向けに作成された新規コードレビューの問題の数
- **修正済** 指定された期間内に完了としてマークされたコード レビューの数
- **経過期間** 作成者がコード レビューの問題を修正するまでに経過した日数

レポートのフィルター/パラメーターの調整

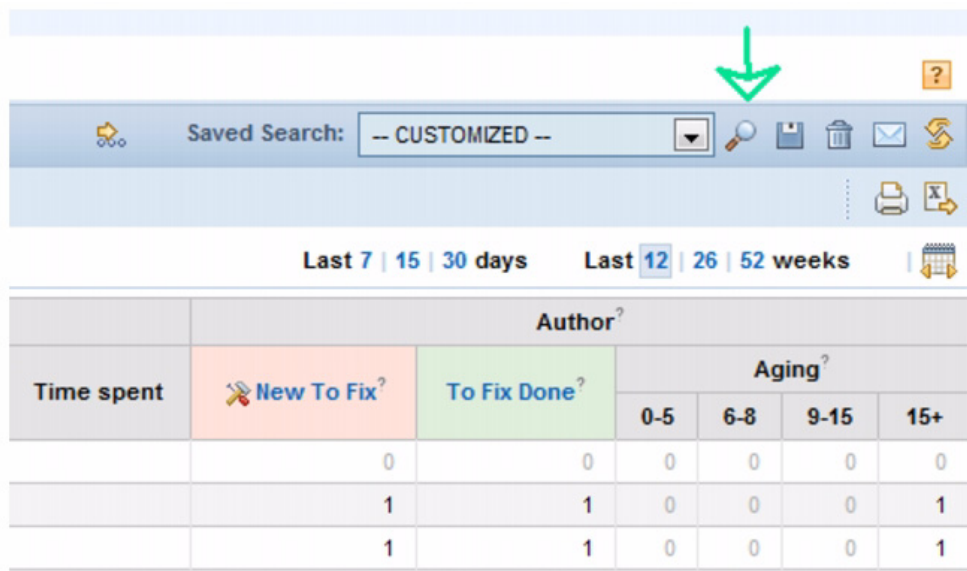
以下のレポート パラメーターを調整すると、簡単に統計値をフィルターできます。

- 時間範囲をクリックして別の期間を参照します。



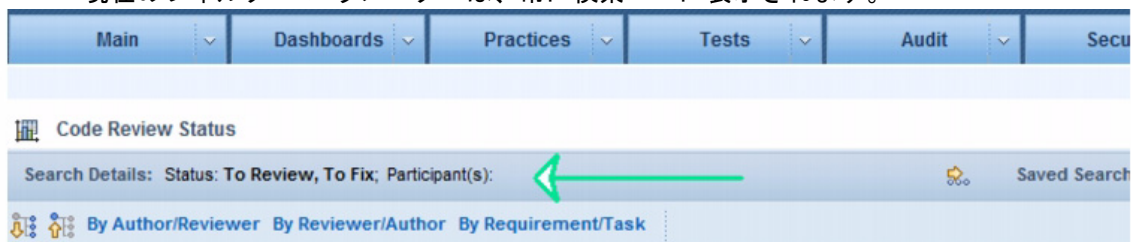
		Author?						
		Time spent	New To Fix?	To Fix Done?	Aging?			
5	15+				0-5	6-8	9-15	15+
0	50		0	0	0	0	0	0
0	9		1	1	0	0	0	1
0	59		1	1	0	0	0	1

- 検索フィルターを開き、選択された期間から特定のコード レビュー データを検索します (たとえば、特定の作成者のコードだけ、または重要度の高い問題だけなど)。



Time spent	Author?		Aging?			
	New To Fix?	To Fix Done?	0-5	6-8	9-15	15+
	0	0	0	0	0	0
	1	1	0	0	0	1
	1	1	0	0	0	1

- 現在のフィルター パラメーターは、常に検索バーに表示されます。



Code Review Status

Search Details: Status: To Review, To Fix; Participant(s):

By Author/Reviewer By Reviewer/Author By Requirement/Task

デフォルトでは、セッションタグを含め (Development Testing Platform 管理のプロジェクト構成ページにあるプロジェクトのコード レビュー フィルターで指定されている場合)、プロジェクトのチーム メンバーの統計値が表示されます。

- プロジェクトのチーム メンバーを構成する方法については、605 ページの「プロジェクトへのユーザーの追加」を参照してください。
- プロジェクトのコード レビュー フィルターを構成する方法については、180 ページの「プロジェクトの作成と構成」を参照してください。

データをレビューアー / 作成者、作成者 / レビューアー、または要求 / タスクごとにグループ化できます。開発者名またはレビューアー名をクリックすると、該当する開発者 / レビューアーのペアの [コード レビュー アクティビティの詳細] レポートが表示されます。詳細については 71 ページの「コード レビューの詳細」を参照してください。

コード レビュー ステータス

このレポートは、選択されたプロジェクトおよび日付のコード レビューのステータスを表示します。これには、保留中のコード レビューの数や、修正すべきコード レビューの問題が含まれます。

Code Review Status

Search Details: Review modified from: 2012-11-03; Review modified to: 2012-11-13; Saved Search: -- CUSTOMIZED --

By Author/Reviewer By Reviewer/Author By Requirement/Task

Author ^?	Reviewer?	To Review?	To Fix?
baranov	jakubiak	18	0
benken	bmerdian	0	0
> cturek	magda, pawelf	10484	0
daniel	riaamour	10	0
griqorb	jakubiak	23	0
> humphrey	mestrada, miove	8	0
jakubiak	tsaucedo	27	0
januszst	pawelf	1	0
iez	jakubiak	0	0
> imadru	benken, jakubiak	24	2
> koczis	pstrycha, rozenau	31	2
kuba	franczak	5	0
marcin	kuba	0	0

デフォルトのビューの各行は、以下のコード レビュー ステータスを作成者 / レビューアーのペアごとに表示します。

- レビューアー - 特定のレビューアーが完了する必要があるコード レビューの数
- 修正 - 特定の作成者が修正する必要があるコード レビューの問題の数

コード レビュー アクティビティ レポートと同様の方法で、レポート データのフィルターおよびグループ化を行うことができます (詳細については 68 ページの「コード レビュー アクティビティ」を参照)。

コード レビューの詳細

[コード レビューの詳細] レポートは、指定されたフィルター (デフォルトでは選択されたプロジェクト) のコード レビュー プロセスの詳細を表示します。以下の情報が含まれます。

- コード レビュー プロセスの参加者
- レビュー対象のソース コード ファイル
- 見つかった問題
- プロセス関係者間で交換されたメッセージ

Code Review Details

Search Details: Status: To Review; Author(s): tsaucedo; Reviewer(s): jakubiak; Review modified from: 2012-11-... Saved Search: -- CUSTOMIZED --

Tasks Authors Reviewers Files Revisions Threads

Summary of Code Review Status: ?

To Review: 21
To Review Done: 0
To Fix: 0
To Fix Done: 0

Details of Code Review Process (status for today): ?

- Items belonging to no task 21 | 0 0 | 0
- tsaucedo 21 | 0 0 | 0
 - jakubiak 21 | 0 0 | 0
 - tests/loadtester/enterprise/reports/BrowserTestReportTest/BrowserTestReportTestForSingleRequest.it 1 | 0 0 | 0
 - webtool/browsertest/BrowserTestingTool.java 1 | 0 0 | 0
 - webtool/browsertest/BrowserTestingToolFailureUtil.java 1 | 0 0 | 0
 - webtool/browsertest/BrowserTestingToolFailureUtilTest.java 1 | 0 0 | 0
 - webtool/report/loadtest/BrowserTestReport.java 2 | 0 0 | 0
 - webtool/report/loadtest/TransactionTestReportContainer.java 1 | 0 0 | 0
 - webtool/report/loadtest/test/BrowserTestReportTest.java 1 | 0 0 | 0
 - xtest/build/NightBuild/et-common-code.iss 1 | 0 0 | 0
 - xtest/build/NightBuild/et.bat 1 | 0 0 | 0
 - xtest/build/NightBuild/et.iss 1 | 0 0 | 0
 - xtest/build/NightBuild/etplugins.iss 1 | 0 0 | 0
 - xtest/build/NightBuild/etvirt.iss 3 | 0 0 | 0
 - xtest/common/com.parasoft.xtest.common.web/src/com/parasoft/xtest/common/web/ui/preference/ImportLegacyPreferencesEditor.java 1 | 0 0 | 0
 - xtest/common/com.parasoft.xtest.common.web/src/com/parasoft/xtest/common/web/ui/preference/Messages.java 1 | 0 0 | 0
 - xtest/common/com.parasoft.xtest.common.web/src/com/parasoft/xtest/common/web/ui/preference/MiscPrefsEditorFactory.java 1 | 0 0 | 0
 - xtest/common/com.parasoft.xtest.common.web/src/com/parasoft/xtest/common/web/ui/preference/res/messages.properties 1 | 0 0 | 0
 - xtest/eclipse/com.parasoft.xtest.common.eclipse.ui.soa/plugin.xml 1 | 0 0 | 0

フィルター ツール バーのカテゴリをクリックすると、データが展開され、次のカテゴリごとに表示されます。

- タスク
- 作成者
- レビューアー
- ファイル リビジョン
- スレッド

[差分] をクリックすると、開発者によって変更され、レビューアーによってレビューされたソースコードを参照できます。

関連項目

以下のマニュアルにもコード レビュー プロセスについての説明があります。

- 315 ページの「コード レビューの統合」
- 『Parasoft Test ユーザーズ ガイド』

[ツールの使用] レポート

[ツールの使用] レポート (図 13) には、ライセンスを要求したユーザーの数、リストされた製品を実行したユーザーの数、製品とプラットフォームごとのライセンス使用状況といった、管理上の目的に役立つ統計が表示されます。

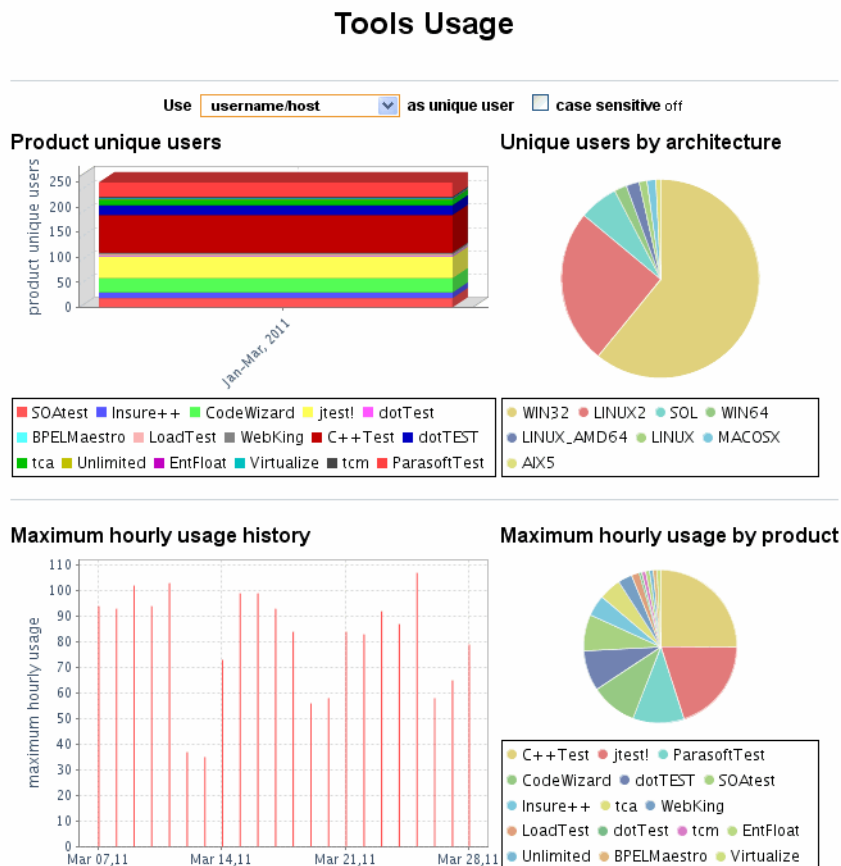
[ツールの使用] のデータは次のレポートに表示されます。

- [製品のユニーク ユーザー] レポート
- [製品別時間ごとの最大使用] グラフ
- [時間ごとの最大使用履歴] レポート
- [アーキテクチャごとのユニーク ユーザー] ページ

ユーザーは、これらのレポートの生成に使用されるデータをカスタマイズすることもできます。詳細については次のセクションで説明します。

- License Server Activity レポートのカスタマイズ

図 13: [ツールの使用] レポート

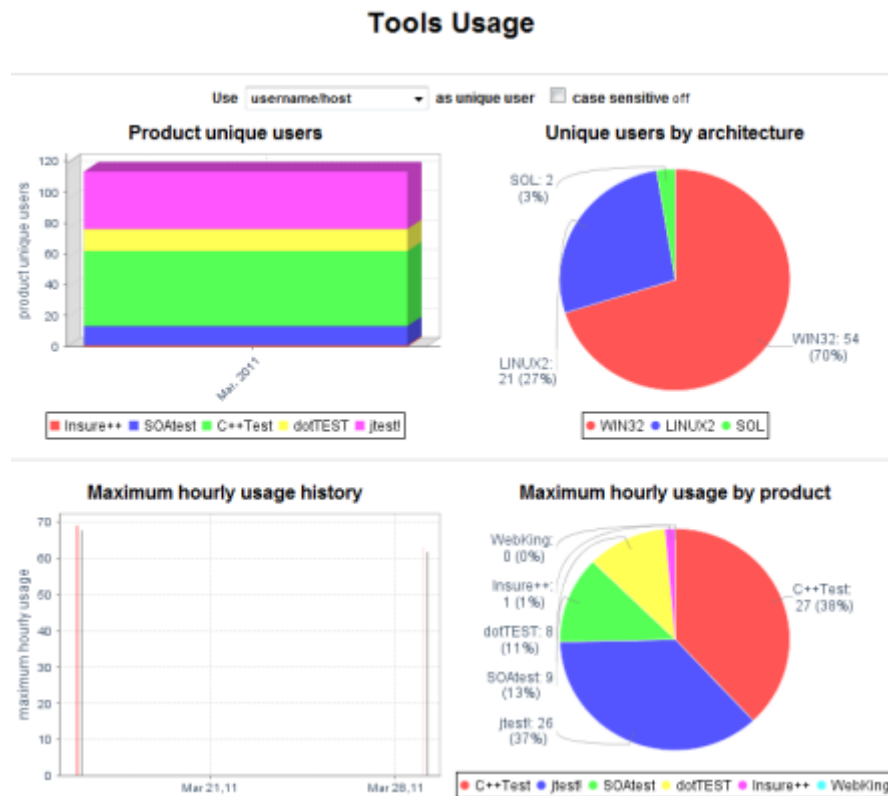


この画面を表示するには : プラクティス > ツールの使用

[製品のユニーク ユーザー] レポート

[製品のユニーク ユーザー] レポートには、指定した期間中にライセンスを要求したユニーク ユーザーの最大数が表示されます。

図 14: [製品のユニーク ユーザー] レポート t



この画面を表示するには：プラクティス > ツールの使用 > 製品のユニーク ユーザー

このレポートには次の情報が表示されます。

- **日** 日次ベースで統計が検証されるようにします。
- **月** 月ベースで統計が検証されるようにします。
- **For** (製品ドロップダウン リスト) データを参照する適切な製品を選択します。
- **Day** ライセンス アクセスが要求された日付。
- **[製品名]** 製品ごとに、対応する日について同時に使用されたライセンスの最大数が表示されます。
- **合計** その日にアクセス要求が発行された、License Server 中の全製品について、最大同時ライセンス数が表示されます。

たとえば、以下の場合には 図 14 に似たレポートが生成されるでしょう。

- USER_1 が午前 8:00 から午前 10:00 まで Jtest ライセンスを使用した。

- USER_2 が午前 9:00 から午前 9:30 まで Jtest ライセンスを使用した。
- USER_1 がまた午後 1:00 から午後 2:00 まで Jtest ライセンスを使用した。
- USER_3 が午後 3:00 から午後 4:00 まで Jtest ライセンスを使用した。

この場合、最大同時ライセンス数は 2 です (SER_1 と USER_2 のライセンス使用が重なったときです)。

まれに次の状況があります。

- USER_1 が 15 分間 (午前 9:00 から午前 9:15) ライセンスを使用して解放した。
- 10 分後、USER_2 (午前 9:25 から午前 11:00) からライセンスが要求された。

この場合、1 時間の間に 9:00 と 9:25 の 2 回ライセンスが要求されたため、同時使用は 2 です。

ライセンス要求の詳細を参照するには、適切な製品と日付の数字をクリックします。License Requests Details ページが表示されます。詳細については「ライセンス要求の詳細」を参照してください。

注意： ユーザー名に加えて、ホスト名もライセンス使用の計測に考慮されます。詳細については 740 ページの「ライセンス サマリ」の Unique Users Recognition Policy の設定についての説明を参照してください。

ライセンス要求の詳細

ライセンス要求の詳細 ページ (図 15) には、表示されている製品と日付について、ライセンス要求の履歴が表示されます。

図 15: ライセンス要求の履歴

日	Insure++	XTEST	合計
2011-08-29-		0	0
2011-04-20-		2	2
2011-04-15-		1	1
2011-04-14-		1	1
2011-04-13-		1	1
2011-04-12-		1	1
2011-03-30		2	2
2011-03-29		1	1

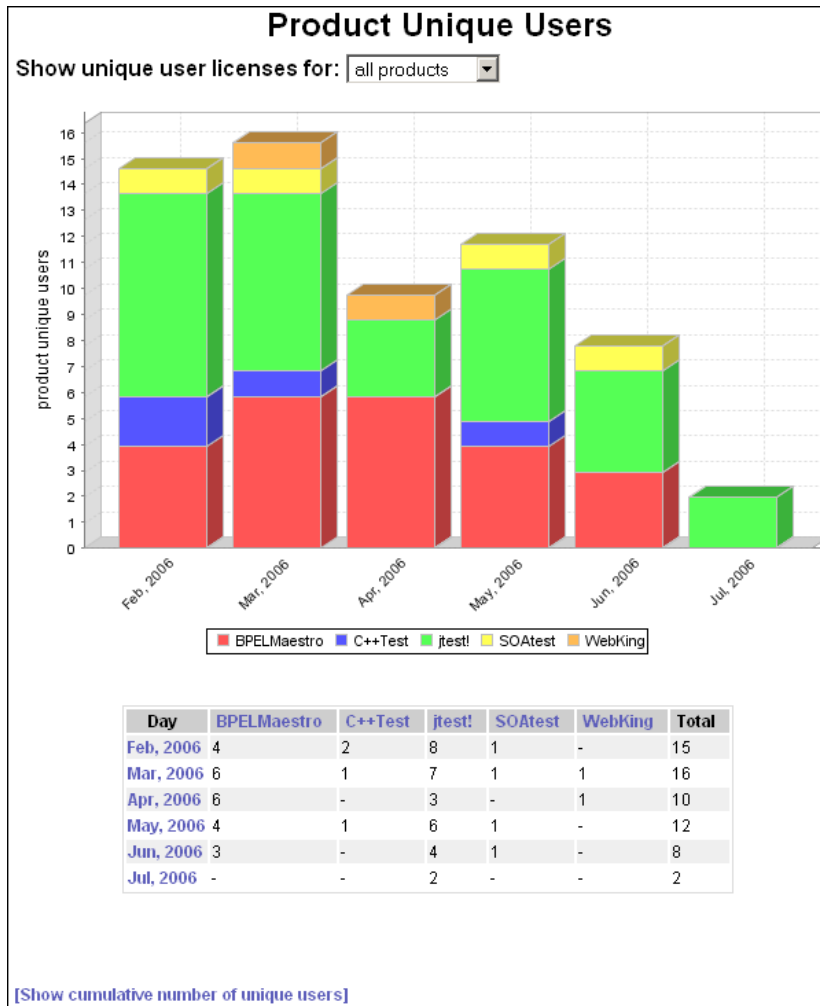
この画面を表示するには：プラクティス > ツールの使用 > 時間ごとの最大使用履歴

[製品別時間ごとの最大使用] グラフ

[製品別時間ごとの最大使用] グラフ (図 16) には、表示されている製品ごとにライセンスを許可されたユニーク ユーザーの総数が表示されます (ユニーク ユーザーは、ユーザー名またマシン ID

によって、またはユーザー名だけによって認識されます)。表示されるデータは月ごとにまとめられるため、棒グラフの各棒はデータの 1 ヶ月分を表します。

図 16: 製品別時間ごとの最大使用



この画面を表示するには：プラクティス > ツールの使用 > 製品別時間ごとの最大使用

[製品別時間ごとの最大使用] ページには、次の情報が表示されます。

- **ユニーク ユーザー ライセンスの表示** ライセンス情報を参照する製品名を選択します。
- **期間** クリックすると、[製品別時間ごとの最大使用の詳細] ページが表示されます。表示されるデータはフィルターされ、選択した月の製品使用が表示されます。詳細については「[製品のユニーク ユーザーの詳細] ページ」を参照してください。
- **< 製品名 >** 対応する月の間に許可された製品ライセンスの数が表示されます。
デフォルトでは、すべての製品のデータが表示されます。データをフィルターするには、製品名をクリックします。選択した製品のデータだけが表示されます。
- **合計** 表示されている日に、選択された製品に対して許可されたライセンスの総数が表示されます。

[製品のユニーク ユーザーの詳細] ページ

[製品のユニーク ユーザーの詳細] ページ (図 17) には、ライセンスの使用データが表示されます。たとえば、どの製品が実行されたか、どのユーザーが実行したか、製品は何日間アクティブだったかなどです。このデータを利用すると、製品が十分に使用されたか、および 1 ヶ月のうちに何個のライセンスが許可されたかを把握できます。また、製品を担当するユーザーが実際に製品を実行しているかどうかも表示されます。

図 17: 製品のユニーク ユーザーの詳細

製品のユニーク ユーザーの詳細

使用 ユーザー名 をユニークユーザーとする 大文字と小文字を区別

Jan-Mar, 2011

ユーザー	XTEST
atsu	1
atuk	5
chie	4
kay	3
shim	3
tmx	6
tmx-se	9
toch	1
総日数:	32
ユーザーの数:	8

製品の列は、特定の製品が特定のユーザーによって各月に何日実行されたかを示します

[CSV形式の結果]

この画面を表示するには: プラクティス > ツールの使用 > 製品のユニーク ユーザー > [日付]

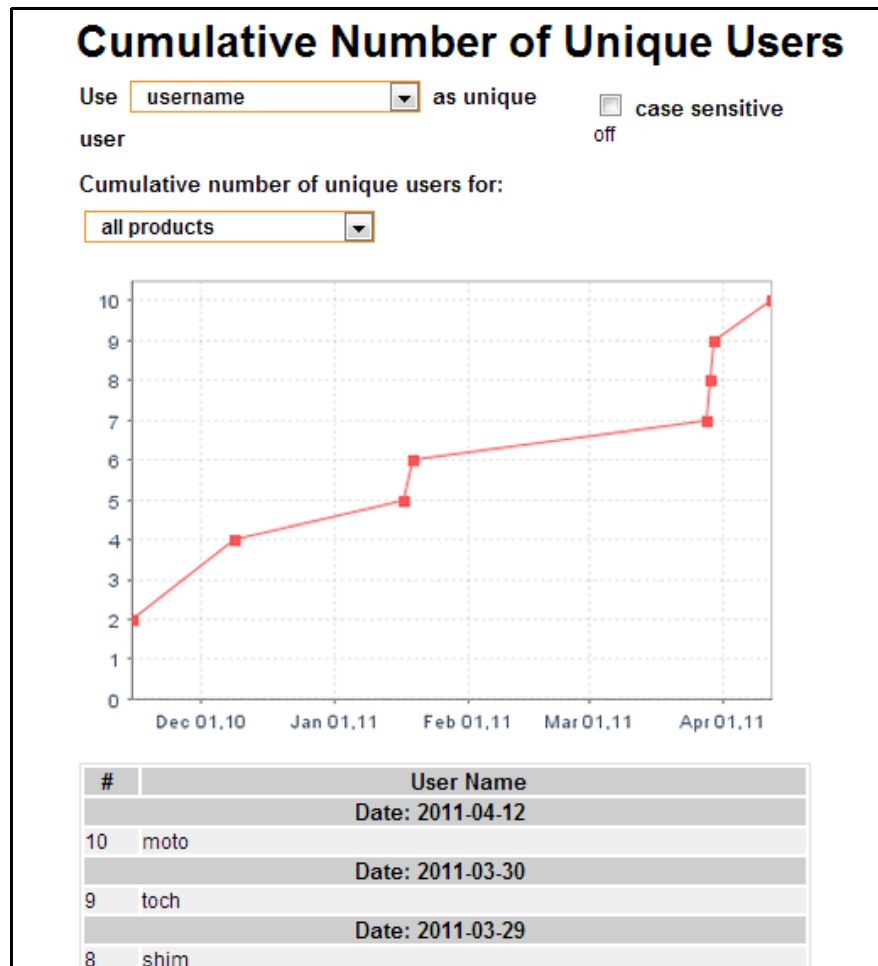
[製品のユニーク ユーザーの詳細] ページには次の情報が表示されます。

- **ユーザー** 製品を実行したユーザーの名前。
- **< 製品名 >** 使用された製品の名前。
- **総日数** 表示されている月において製品がアクティブだった日数。
- **ユーザーの数** 製品を実行したユーザーの数。
- **[CSV形式の結果]** このレポートの結果をカンマ区切りファイルとしてダウンロードし、特定のユーザーのシステムに Excel などアップロードすることができます。

[ユニーク ユーザー数の累積を表示] ページ

[ユニーク ユーザー数の累積を表示] ページには、すべての製品または選択された製品についてユニーク ユーザーの累積統計が表示されます。デフォルトは「すべての製品」です。

図 18: [ユニーク ユーザー数の累積を表示] ページ



この画面を表示するには：プラクティス > ツールの使用 > 製品のユニーク ユーザー > [ユニーク ユーザー数の累積を表示]

[ユニーク ユーザー数の累積を表示] ページには次の情報が表示されます。

- **#** 製品が実行された回数。
- **ユーザー名** 製品を実行したユーザーの名前。
- **マシン名** 製品が実行されたマシンの名前。
- **日付** 製品が実行された日付。

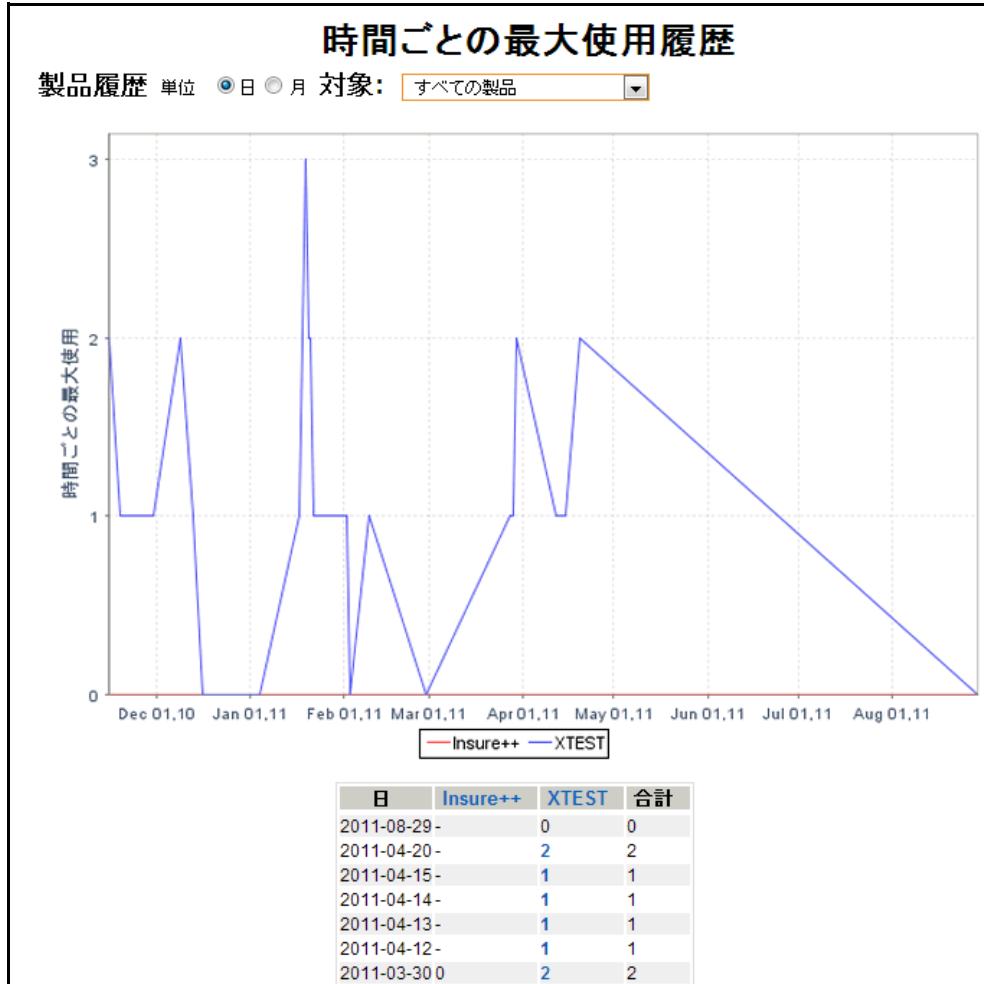
[時間ごとの最大使用履歴] レポート

[時間ごとの最大使用履歴] レポートは、[製品別時間ごとの最大使用] レポートと同じデータを使用します。ただし、データが日付ではなく製品でソートされます。

[ツールの使用] レポート (図 13) の [製品別時間ごとの最大使用] の円グラフで、履歴を参照したい製品をクリックします。[時間ごとの最大使用履歴] レポート (図 19) が表示され、選択した製品

について許可されたライセンスの統計を参照することができます。このレポートに表示されるデータの詳細については 75 ページの「[製品別時間ごとの最大使用] グラフ」を参照してください。

図 19: [時間ごとの最大使用履歴] レポート

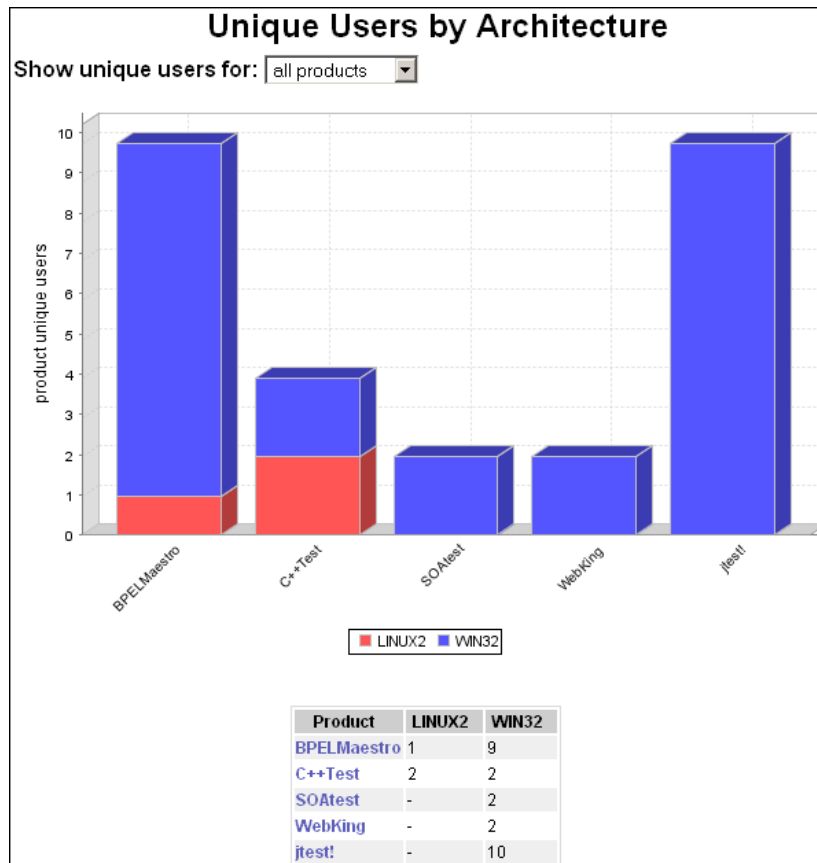


この画面を表示するには: プラクティス > ツールの使用 > 時間ごとの最大使用履歴

[アーキテクチャごとのユニークユーザー] ページ

[アーキテクチャごとのユニークユーザー] ページ (図 20) には、オペレーティング システム アーキテクチャ (Windows または Linux) に基づいて、Parasoft 製品を実行中のユニークユーザーの数が表示されます。

図 20: [アーキテクチャごとのユニーク ユーザー] ページ



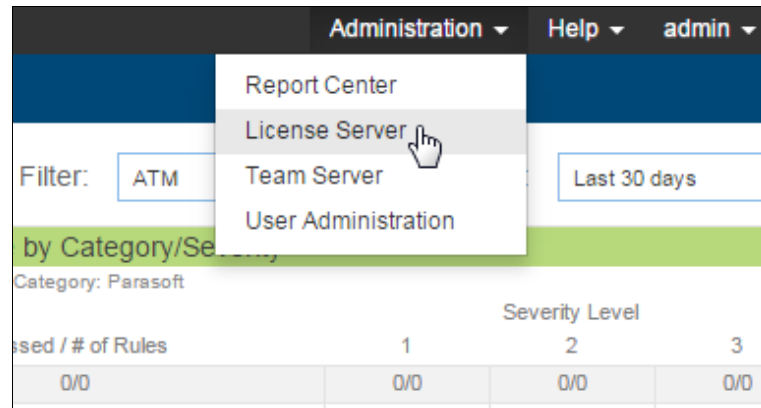
この画面を表示するには : プラクティス > ツールの使用 > アーキテクチャごとのユニーク ユーザー

[アーキテクチャごとのユニーク ユーザー] ページには次の情報が表示されます。

- **製品** 実行された製品の名前。特定の製品名をクリックすると、その製品でレポートがフィルターされます。
- **LINUX2** このアーキテクチャで製品を実行したユーザーの数。
- **WIN32** このアーキテクチャで製品を実行したユーザーの数。


License Server Activity レポートのカスタマイズ

ツールの使用 (License Server Activity) レポートに表示するデータは、License Server 管理者ページからカスタマイズすることができます。[管理] ドロップダウン メニューから [License Server] を選択します。



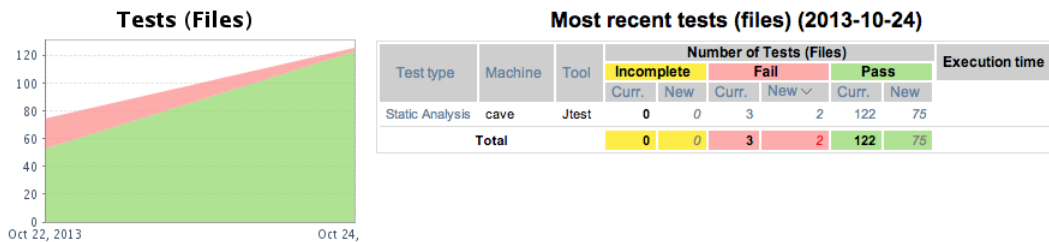
利用できる設定の詳細については 208 ページの「License Server アクティビティ レポートの設定」を参照してください。

[テストの概要] レポート

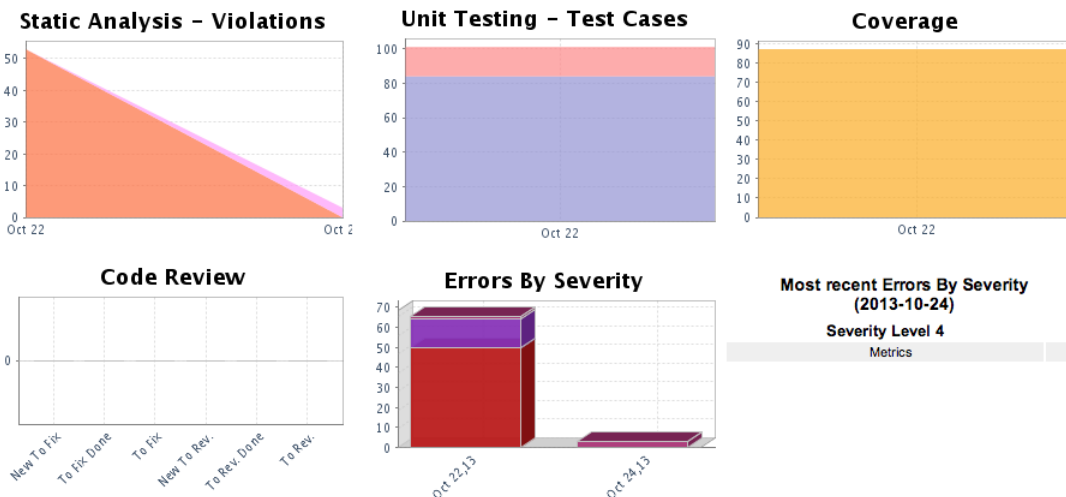
レポート ビューで [テスト] メニューの [テストの概要] をクリックすると、テストの概要ページが開きます。このページには、指定の期間中に実行された、選択された製品のすべてのテスト結果が表示されます。テスト結果の全体的な傾向だけでなく、最後に実行されたテストの詳細もレポートされます。[テストの概要] セクション () には、テスト結果の全体的な傾向が表示されます。静的解析、単体テスト、機能テストなど、選択されたプロジェクトについて実行されたすべての自動テストが含まれます。

Last 7 | 15 | 30 days

Tests Overview



Tests Overview Details



Current Tasks for Developers

Last Generation Time: N/A

Summary report

Last Run: N/A
 There were no errors reported for the project.

グラフは、単体テストと静的解析で異なるデータを別に表示します。

- 単体テスト テスト (スイート) ファイルの数
- 静的解析 テスト済みファイルの数

テスト (ファイル) レポートの詳細については、133 ページの「テスト (ファイル)」を参照してください。

[テストの概要] セクションの [テスト (ファイル)] グラフは、指定された期間のテストの傾向を反映しています。

グラフは、単体テストのデータと静的解析のデータを異なる方法で表示します。

- 単体テストの場合、テスト（テストスイート）ファイルの数を表示します。
- 静的解析の場合、テスト対象ファイルの数を表示します。

[テスト（ファイル）] レポートの詳細については、133 ページの「テスト（ファイル）」を参照してください。

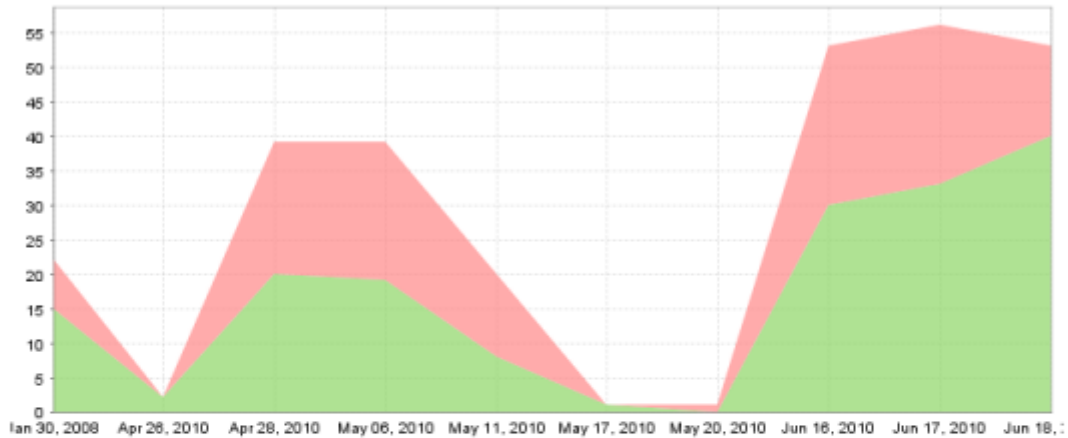
[日付別テスト]レポート

[日付別テスト]レポート(図 21)には、プロジェクトの現在の状態が分かりやすく表示されます。表は、選択された期間に実行されたテストのリストです。各行には、特定の日に実行されたすべてのテストのテストケースについて、未完了、失敗、成功の数が表示されます。このテストには、静的解析、単体テスト、回帰テストなどが含まれます。

詳細については 133 ページの「テスト(ファイル)」を参照してください。

図 21: [日付別テスト]レポート

テスト



日付	未完了	失敗	成功	合計	継続時間	実行時間 Diff
Jun 18, 2010	0	13	40	53	2m 40s	-1m 27s
Jun 17, 2010	0	23	33	56	4m 7s	1m 38s
Jun 16, 2010	0	23	30	53	2m 29s	2m 6s
May 20, 2010	0	1	0	1	23s	18s
May 17, 2010	0	0	1	1	4s	-1m 1s
May 11, 2010	0	12	8	20	1m 5s	-51s
May 06, 2010	0	20	19	39	1m 57s	7s
Apr 28, 2010	0	19	20	39	1m 50s	1m 31s
Apr 26, 2010	0	0	2	2	19s	-1s
Jan 30, 2008	0	7	15	22	20s	

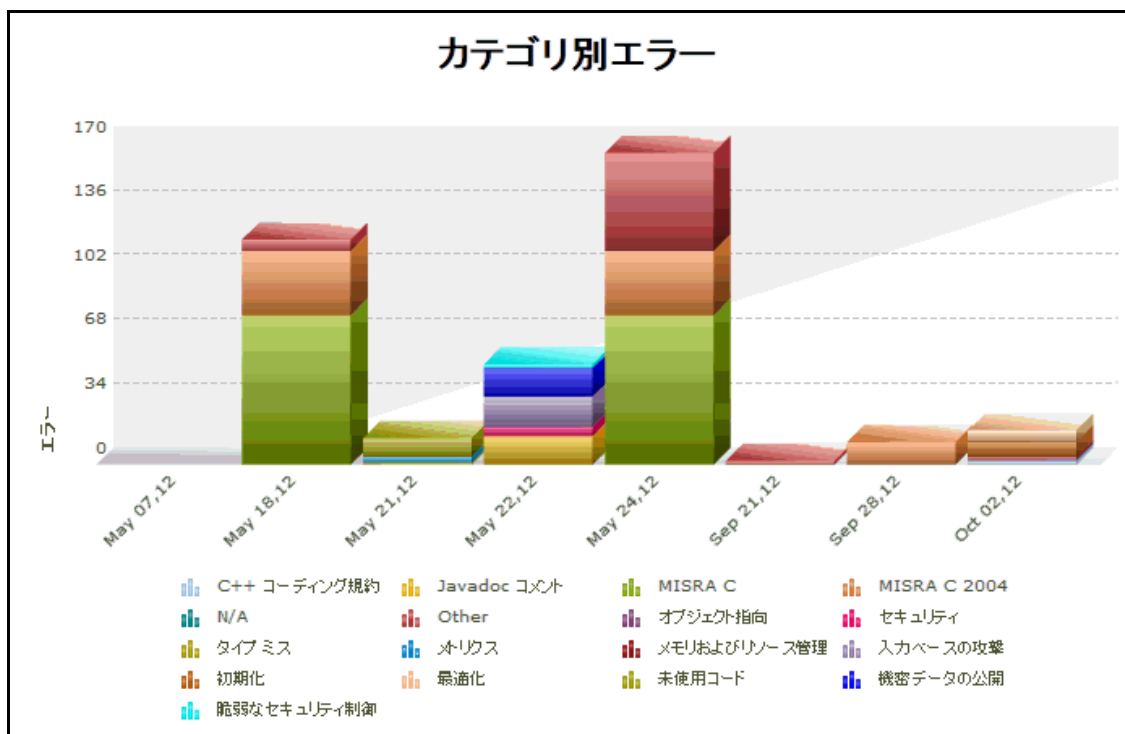
この画面を表示するには: テスト > 日付別テスト

[カテゴリ別エラー] レポート

[カテゴリ別エラー] レポートでは、コード中の静的解析違反と単体テスト エラーがどのように分布しているかをチェックすることができます。エラー カテゴリは、レポートを送信するツールによって異なります。

[カテゴリ別エラー] レポートを表示するには、[テスト] メニューの [カテゴリ別エラー] を選択します。このグラフは、数字ではなく視覚化を利用してエラー カテゴリの割合を表します。エラー カテゴリ (タイプ) ごとに異なる色が割り当てられます。選択されたプロジェクトの選択されたドロップ内の日付ごとに、色分けされたブロックが積み上げられ、各ブロックは、そのプロジェクトのテスト中に発見された静的解析違反と単体テスト エラーの数を反映します。どのエラー タイプが多いかが一目で把握できます。

図 22: [カテゴリ別エラー] レポート



この画面を表示するには : テスト > カテゴリ別エラー

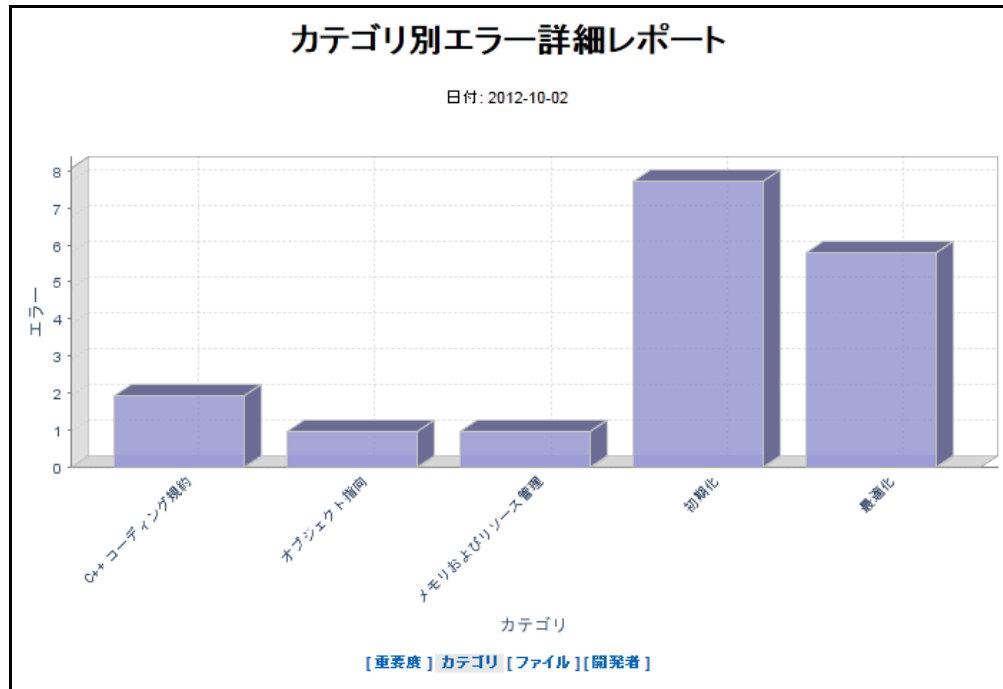
[カテゴリ別エラー] レポートから詳細情報にアクセスする

[カテゴリ別エラー] レポート中で特定の日付をクリックすると、その日付に対する カテゴリ別エラー詳細レポート が表示されます。

カテゴリ別エラー詳細レポート

カテゴリ別エラー詳細レポート (図 23) は、ページの上部に表示され、選択されたドロップ日について、静的解析違反と単体テスト エラーの数がエラー カテゴリ別にソートされて表示されます。棒グラフの各棒は、特定のタイプを表します。オプションバーからドロップ日を変更できます。

図 23: カテゴリ別エラー詳細レポート



この画面を表示するには : テスト > カテゴリ別エラー > [日付] (グラフ)

カテゴリ別エラー詳細レポートの下部の表 (図 24) には、カテゴリがアルファベット順にリストされ、そのカテゴリにおける静的解析違反と単体テスト エラーの数が表示されます。各カテゴリの下には、エラーの数が 1 から 5 の重要度で分類されて表示されます (1 が最も高い重要度を表し、5 は最も低い重要度を表します)。

図 24: カテゴリ別エラー詳細レポートの表

[重要度] [カテゴリ] [ファイル] [開発者]	
C++ コーディング規約	3
Severity Level 1	3
Other	48
Severity Level 3	48
オブジェクト指向	8
Severity Level 2	8
バグの可能性	1
Severity Level 2	1
リソース	3
Severity Level 1	3
最適化	5
Severity Level 3	5

カテゴリ別エラー詳細レポートの表からさらに詳細情報にアクセスするグラフと表の表示を変更するには、次の操作を行います。

- 表の上にあるリンクをクリックすると、別の観点からエラーを参照することができます。
 - 重要度：重要度別エラー詳細レポートが開きます。
 - ファイル：ファイル別エラー詳細レポートが開きます。
 - 開発者：開発者別エラー詳細レポートが開きます。

カテゴリ別にリストされたエラーの詳細を参照するには、次の操作を行います。

- 適切なカテゴリと重要度レベルに対応した統計をクリックします。図 25 に似たページにエラーについてのさらに詳しい情報が表示されます。

図 25: カテゴリ別エラー詳細レポート (さらに詳細な情報)

エラーの詳細レポート

[重要度] [カテゴリ] [ファイル] [開発者]

C++ コーディング規約

Priority Level 1				
ファイル名	開発者	行	エラー	ルール
Check.h	masuya	22	変換を可能にするコンストラクタ CndCheck を explicit にする	CODSTA-CPP-04
Check.h	不明	22	変換を可能にするコンストラクタ CndCheck を explicit にする	CODSTA-CPP-04
utString.h	masuya	29	変換を可能にするコンストラクタ OutputString を explicit にする	CODSTA-CPP-04

この画面を表示するには : テスト > カテゴリ別エラー > [日付] (グラフ) > [エラーの数]

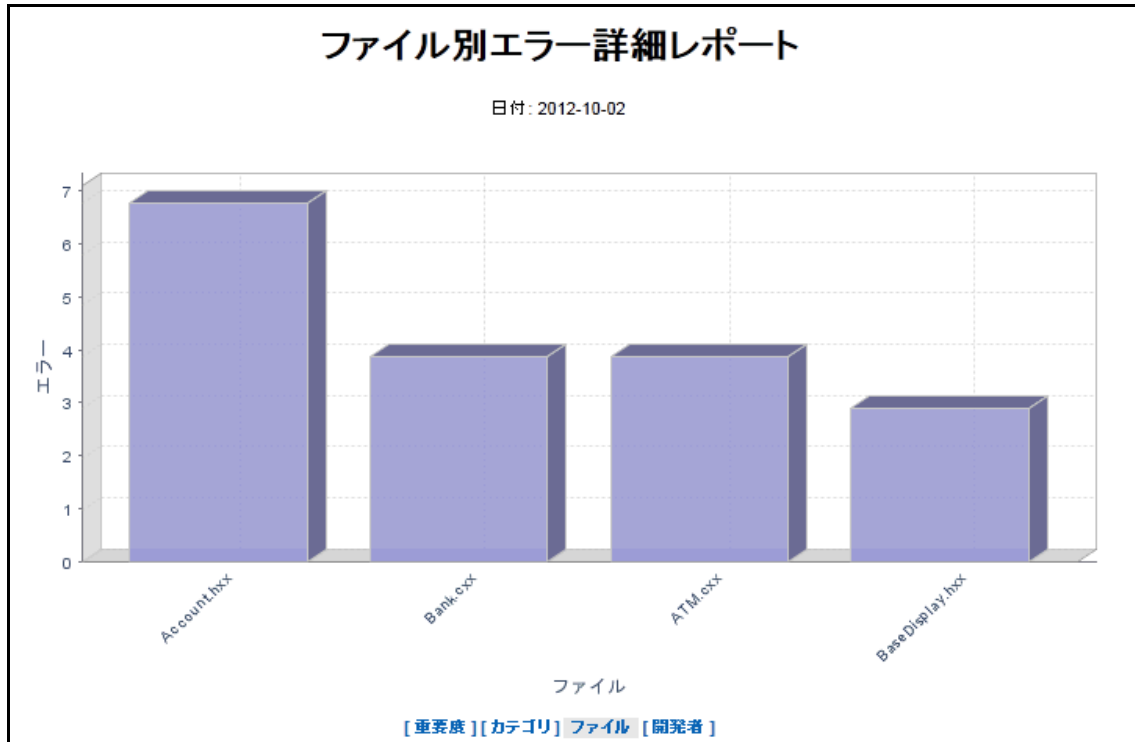
以下は、カテゴリ別エラー詳細レポートの表の列ヘッダーについての説明です。

- ファイル名** エラーがあるファイルの名前。
 この列ヘッダーをクリックすると、ファイル別エラー詳細レポート が開きます。列ヘッダーの下の特定のファイル名をクリックすると、そのソース ファイルが表示されます。
- 開発者** 対応するエラーに責任がある開発者のユーザー名。
 クリックすると、開発者別エラー詳細レポート が開きます。
- 行** エラーが発見されたコードの行番号。
 特定の行をクリックすると、その行が黄色でハイライトされた状態でソース ファイルが開きます。
- エラー** 発生したエラーについての説明。
- ルール** 対応するエラーで違反があったルールの名前。
 特定のルールをクリックすると、そのルールの説明が表示されます。加えて、関連する注意事項、セキュリティとの関連、パラメーター、利点、デメリット、サンプル コード、修正方法、および関連資料についての情報も表示されます。

ファイル別エラー詳細レポート

ファイル別エラー詳細レポート (図 26) には、エラーの多い上位 10 ファイルが表示されます。

図 26: ファイル別エラー詳細レポート



この画面を表示するには:

- オプション 1: テスト > カテゴリ別エラー > [日付] (グラフ) > [ファイル] リンク
- オプション 2: テスト > 重要度別エラー > [日付] (グラフ) > [ファイル]

ファイル別エラー詳細レポートの表 (図 27) には、エラーが多い上位 10 ファイルのリストと各ファイルにおけるエラーの総数が表示されます。

図 27: ファイル別エラー詳細レポートの表

[重要度] [カテゴリ]	ファイル	[開発者]
	TestSuite_KeywordString_cpp.cpp	16
	TestSuite_CmdCheck_cpp.cpp	14
	Outputcli.cpp	8
	TestSuite_CommentString_cpp.cpp	5
	TestSuite_SrcCodeString_cpp.cpp	4
	OutputString.h	3
	CmdCheck.cpp	3
	KeywordString.h	2
	OutputLine.cpp	2
	CommentString.h	2

ファイル別エラー詳細レポートの表からさらに詳細情報にアクセスする

グラフと表の表示を変更するには、次の操作を行います。

- 表の上にあるリンクをクリックすると、別の観点からエラーを参照することができます。
 - 重要度：重要度別エラー詳細レポートが開きます。
 - カテゴリ：カテゴリ別エラー詳細レポートが開きます。
 - 開発者：開発者別エラー詳細レポートが開きます。

ファイルのソースコードを参照するには、次の操作を行います。

- リンクがある場合、適切なファイル名をクリックします。

ファイル別にリストされたエラーの詳細をさらに参照するには、次の操作を行います。

- 適切なファイル名に対応したエラーの数をクリックします。図 28 に似たページが開き、ファイルのエラーについてさらに詳しい情報が表示されます。

図 28: ファイル別エラー詳細レポート (さらに詳細な情報)

エラー詳細レポート

[重要度] [カテゴリ] **ファイル** [開発者]

TestSuite_CommentString_cpp.cpp

開発者	行	エラー	ルール
masuya	321	テスト ケースを実行中に標準例外が発生しました: vector<T> too long	N/A
masuya	378	テスト ケースを実行中に標準例外が発生しました: vector<T> too long	N/A
masuya	435	テスト ケースを実行中に標準例外が発生しました: vector<T> too long	N/A
masuya	476	テスト ケースを実行中に標準例外が発生しました: vector<T> too long	N/A
masuya	566	テスト ケースを実行中に標準例外が発生しました: vector<T> too long	N/A

この画面を表示するには:

オプション 1: テスト > カテゴリ別エラー > [日付] (グラフ) > [ファイル] > [エラーの数] (表)

オプション 2: テスト > 重要度別エラー > [日付] (グラフ) > [ファイル] [エラーの数] (表)

以下は、ファイル別エラー詳細レポートの表の列ヘッダーについての説明です。

- 開発者** 対応するエラーに責任がある開発者のユーザー名。
クリックすると、開発者別エラー詳細レポートが開きます。
- ファイル名** エラーがあるファイルの名前。
このヘッダーをクリックすると、ファイル別エラー詳細レポートが開きます。ヘッダーの下のファイル名をクリックすると、そのソースファイルが表示されます。
- 行** エラーがあったコードの行番号。

特定の行をクリックすると、その行が黄色でハイライトされた状態でソース ファイルが開きます。

- **エラー** 発生したエラーについての説明。
- **ルール** 対応するエラーで違反があったルールの名前。

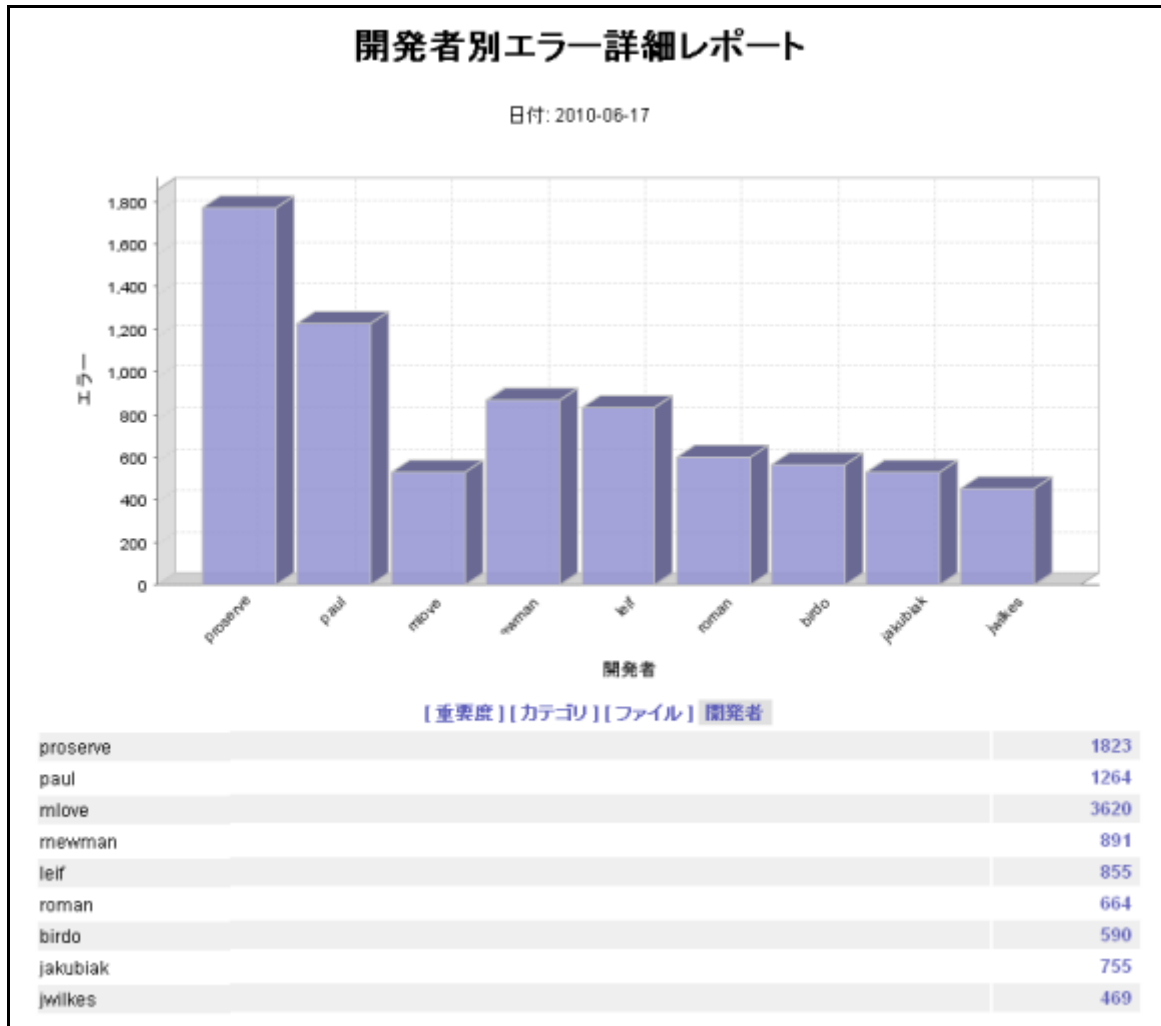
特定のルールをクリックすると、そのルールの説明が表示されます。加えて、関連する注意事項、セキュリティとの関連、パラメーター、利点、デメリット、サンプル コード、修正方法、および関連資料についての情報も表示されます。

開発者別エラー詳細レポート

開発者別エラー詳細レポート (図 29) には、エラーが多い上位 10 人の開発者が表示されます。

開発者別エラー詳細レポートの下表には、検出された静的解析違反と単体テスト エラーに責任がある、すべての開発者が表示されます。各開発者のエラーの総数も表示されます。

図 29: 開発者別エラー詳細レポート



この画面を表示するには：

- オプション 1: テスト > カテゴリ別エラー > [日付] (グラフ) > [開発者]
- オプション 2: テスト > 重要度別エラー > [日付] (グラフ) > [開発者]
- オプション 3: ファイル別エラー詳細レポート > [エラーの数] (グラフ) > [開発者]

開発者別エラー詳細レポートの表からさらに詳細情報にアクセスする

グラフと表の表示を変更するには、次の操作を行います。

- 表の上にあるリンクをクリックすると、別の観点からエラーを参照することができます。
 - 重要度：重要度別エラー詳細レポート が開きます。
 - カテゴリ：カテゴリ別エラー詳細レポート が開きます。
 - ファイル：ファイル別エラー詳細レポート が開きます。

開発者ごとにリストされたエラーについての詳細を参照するには、次の操作を行います。

- 開発者名に対応するエラーの数をクリックすると、図 25 に似たページが開き、その開発者のエラーについてさらに詳しい情報が表示されます。

図 30: 開発者別エラー詳細レポート (さらに詳細な情報)

エラー詳細レポート			
[重要度] [カテゴリ] [ファイル] [開発者]			
masuya			
ファイル名	行	エラー	ルール
CommentString.cpp	27	The parameter 'strLine' of function 'CommentString' is passed by value	OPT-14
CommentString.h	17	クラス CommentString に仮想関数があるが、仮想デストラクタがない	OOP-23
KeywordString.h	11	クラス KeywordString に仮想関数があるが、仮想デストラクタがない	OOP-23
OutputLine.cpp	23	The parameter 'strLine' of function 'OutputLine' is passed by value	OPT-14
OutputString.h	17	クラス OutputString に仮想関数があるが、仮想デストラクタがない	OOP-23
OutputString.h	29	実装を可能にするコンストラクタ OutputString を explicit にする	CODSTA-CPP-04
Outputcli.cpp	30	アクセス違反の例外 [DPPTTEST_ACCESS_VIOLATION]	N/A
Outputcli.cpp	30	アクセス違反の例外 [DPPTTEST_ACCESS_VIOLATION]	N/A
Outputcli.cpp	57	"pOadCheck" が null チェックより前に間接参照されている : .C Outputcli.cpp (56): pOadCheck->Output(); .P Outputcli.cpp (57): if (pOadCheck != NULL) // 重要な理: pOadCheck != null	BD-PB-DEREF
Outputcli.cpp	76	アクセス違反の例外 [DPPTTEST_ACCESS_VIOLATION]	N/A
Outputcli.cpp	76	アクセス違反の例外 [DPPTTEST_ACCESS_VIOLATION]	N/A
Outputcli.cpp	76	アクセス違反の例外 [DPPTTEST_ACCESS_VIOLATION]	N/A

この画面を表示するには：

- オプション 1: テスト > カテゴリ別エラー > [日付] (グラフ) > [開発者] > [エラーの数]
- オプション 2: テスト > 重要度別エラー > [日付] (グラフ) > [開発者] > [エラーの数]
- オプション 3: ファイル別エラー詳細レポート > [エラーの数] (グラフ) > [開発者] > [エラーの数]

以下は、開発者別エラー詳細レポートの表の列ヘッダーについての説明です。

- **ファイル名** エラーがあるファイルの名前。
このヘッダーをクリックすると、ファイル別エラー詳細レポートが開きます。ヘッダーの下のファイル名をクリックすると、そのソース ファイルが表示されます。
- **行** エラーがあったコードの行番号。
特定の行をクリックすると、その行が黄色でハイライトされた状態でソース ファイルが開きます。
- **エラー** 発生したエラーについての説明。
- **ルール** 対応するエラーで違反があったルールの名前。
特定のルールをクリックすると、そのルールの説明が表示されます。加えて、関連する注意事項、セキュリティとの関連、パラメーター、利点、デメリット、サンプル コード、修正方法、および関連資料についての情報も表示されます。

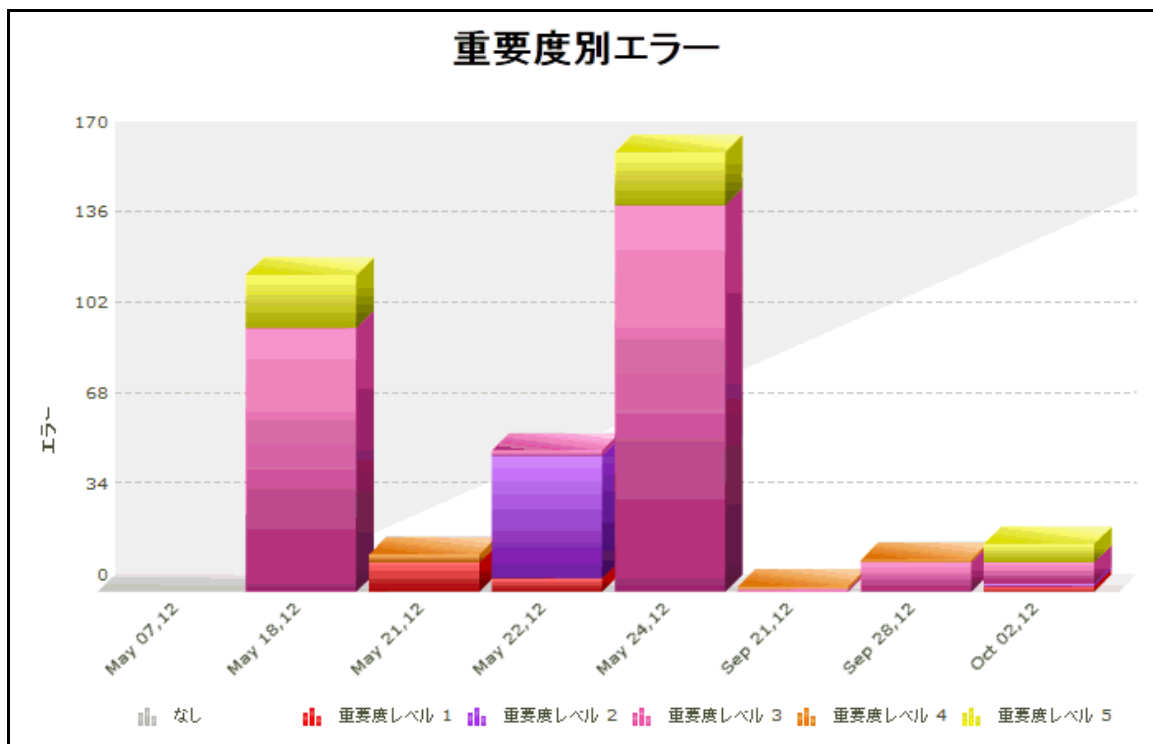
[重要度別エラー]レポート

[重要度別エラー]レポートを開くには、[テスト]メニューの[重要度別エラー]を選択します。このグラフは、数字ではなく視覚化を利用して、異なる静的解析違反と単体テストエラーの割合を表します。重要度レベルごとに異なる色が割り当てられます。重要度レベルを決定するのはユーザーです。ただし通常、重要度レベルは1から5の範囲であり、1が最も重要度が高く、最も心配されるエラーです。

選択されたプロジェクトの選択されたドロップ内の日付ごとに、色分けされたブロックが積み上げられ、各ブロックは、そのプロジェクトのテスト中に発見された静的解析違反と単体テストエラーの数を反映します。どの重要度レベルが多いかが一目で把握できます。

[重要度別エラー]レポートは、最も重要度が高い静的解析違反と単体テストエラーが解決されたかどうかを確認するための有効な手段でもあります。[重要度別エラー]レポートのグラフの棒の高さは、エラーが解決されているかを反映して、日増しに低くなるべきです。

図 31: [重要度別エラー]レポート



この画面を表示するには：テスト > 重要度別エラー

[重要度別エラー]レポートから詳細情報にアクセスする

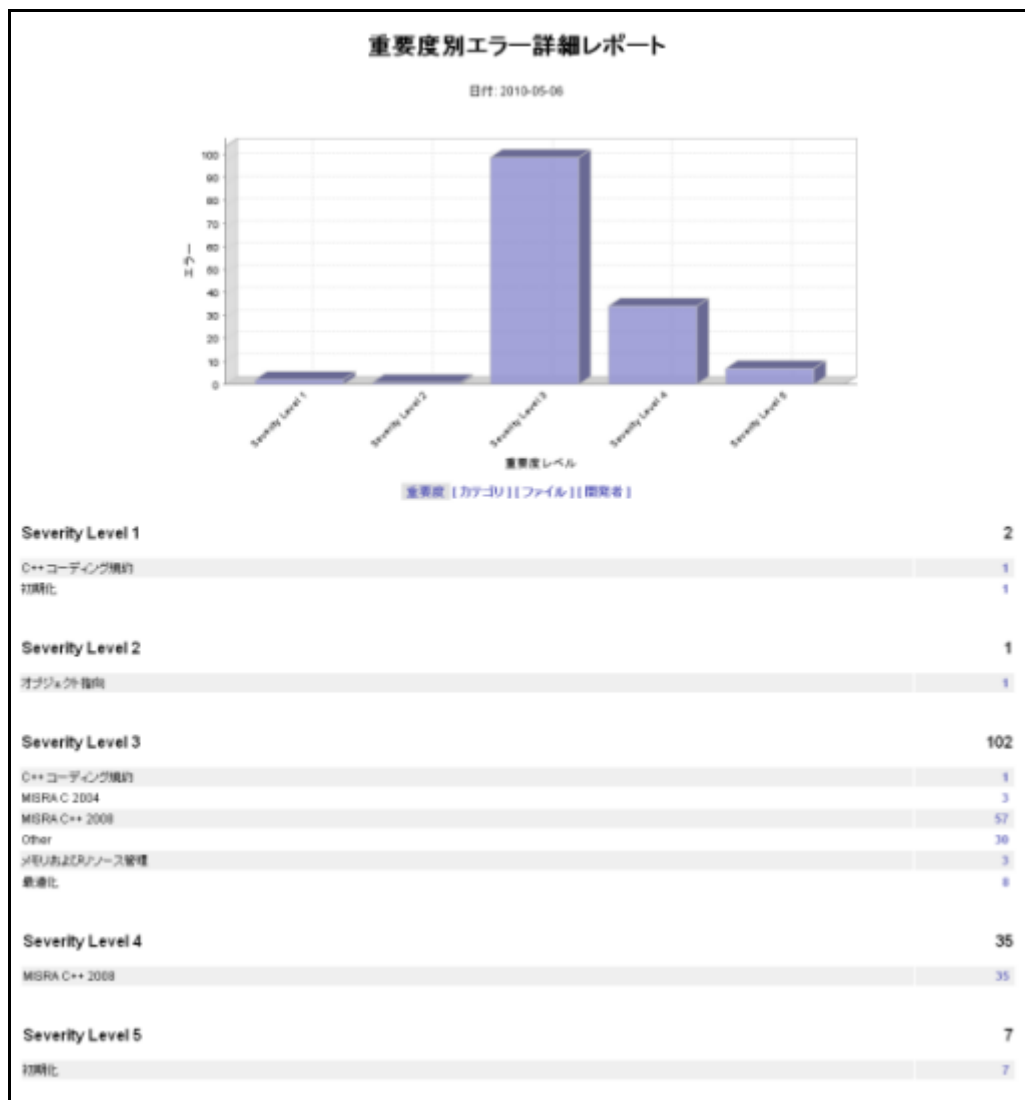
[重要度別エラー]レポート中の特定の日付をクリックすると、重要度別エラー詳細レポートが開きます。

重要度別エラー詳細レポート

重要度別エラー詳細レポートには、選択されたドロップ日について、重要度レベルでソートされた静的解析違反と単体テストエラーの数が表示されます。たとえば1から5の5レベルの重要度がある場合(1が最も重要度が高く、5が最も低い)、図32のようなグラフが表示されます。各棒が異なる重要度レベルを表します。

重要度別エラー詳細レポートの下の表には、検出された静的解析違反と単体テストエラーが重要度レベル別にリストされます。選択したテスト実行日について、検出された静的解析違反と単体テストエラーの総数が重要度レベルごとに表示されます。各重要度レベルの下には、静的解析違反と単体テストエラーが検出されたすべてのルールカテゴリと、それぞれのカテゴリで発見されたエラーの数が表示されます。

図 32: 重要度別エラー詳細レポート



この画面を表示するには: テスト > 重要度別エラー > [日付] (グラフ)

重要度別エラー詳細レポートの表から詳細情報にアクセスする

グラフと表のビューを変更するには、次の操作を行います。

- 表の上にあるリンクをクリックすると、別の観点からエラーを参照することができます。
 - カテゴリ：カテゴリ別エラー詳細レポートが開きます。
 - ファイル：ファイル別エラー詳細レポートが開きます。
 - 開発者：開発者別エラー詳細レポートが開きます。

重要度別にリストされたエラーの詳細をさらに参照するには、次の操作を行います。

- 適切なルールカテゴリに対応したエラーの数をクリックします。図 33 に似たページが開き、エラーについてさらに詳しい情報が表示されます。

図 33: 重要度別エラーの詳細レポート (さらに詳細な情報)

エラーの詳細レポート				
重要度 [カテゴリ][ファイル][開発者]				
Severity Level 3				
最適化				
ファイル名	開発者	行	エラー	ルール
ATM.cxx	-	10	The parameter 'password' of function 'viewAccount' is passed by value	OPT-14
Account.hxx	-	11	クラス Account はインライン コンストラクタを定義している	OPT-17
Account.hxx	-	16	クラス Account はインライン コンストラクタを定義している	OPT-17
Bank.cxx	-	15	The parameter 'password' of function 'getAccount' is passed by value	OPT-14
BaseDisplay.hxx	-	9	クラス BaseDisplay はインライン コンストラクタを定義している	OPT-17
BaseDisplay.hxx	-	10	クラス BaseDisplay はデストラクタをインラインとして定義している	OPT-17
TestObjectFactory.hxx	-	5	クラス TestObjectFactory はインライン コンストラクタを定義している	OPT-17
TestObjectFactory.hxx	-	7	クラス TestObjectFactory はデストラクタをインラインとして定義している	OPT-17

この画面を表示するには：テスト > 重要度別エラー > [日付] (グラフ) > [エラーの数]

以下は、重要度別エラー詳細レポート (さらに詳細な情報) の表の列ヘッダーについての説明です。

- ファイル名** エラーがあるファイルの名前。
このヘッダーをクリックすると、ファイル別エラー詳細レポートが表示されます。このヘッダーの下にファイル名をクリックすると、そのソースファイルが表示されます。
- 開発者** 対応するエラーに責任がある開発者の名前。
このヘッダーをクリックすると、開発者別エラー詳細レポートが表示されます。
- 行** エラーがあったコードの行番号。
特定の行をクリックすると、その行が黄色でハイライトされた状態でソースファイルが開きます。
- エラー** 発生したエラーについての説明。
- ルール** 対応するエラーで違反があったルールの名前。

特定のルールをクリックすると、そのルールの説明が表示されます。加えて、関連する注意事項、セキュリティとの関連、パラメーター、利点、デメリット、サンプルコード、修正方法、および関連資料についての情報も表示されます。

[最近のテスト ログ] レポート

[最近のテスト ログ] レポート (図 34) には、ツールによって Report Center データベースに正常に送信されたすべてのログが、選択されたプロジェクトと期間について表示されます。Report Center データベースによるログが正常に取得されたかどうかが表示されるほか、管理者、QA チームメンバー、および Report Center へのログ送信を担当する指定チームのための統計が表示されます。ログが正常に取得されなかった場合は、Report Center ツールの構成を確認して再びテストを実行するべきです。

図 34: [最近のテスト ログ] レポート

最近のテスト ログ								
44 発見された行								
ログID	日付	未完了のテスト	失敗したテスト	成功したテスト	ツール	ユーザー	マシン	テスト合計
2502	Jun 18, 2010 7:50:24 PM	0	0	0	1 SourceScanner	masuya	dell-83800-22.tmx.local	1
2254	Jun 18, 2010 6:52:41 PM	0	1	14	C++test	masuya	dell-83800-22	15
2253	Jun 18, 2010 6:22:34 PM	0	4	11	C++test	masuya	dell-83800-22	15
2252	Jun 18, 2010 6:20:09 PM	0	5	10	C++test	masuya	dell-83800-22	15
2002	Jun 18, 2010 5:59:56 PM	0	3	4	C++test	masuya	dell-83800-22	7
1765	Jun 17, 2010 9:30:00 PM	0	0	0	1 SourceScanner	masuya	dell-83800-22.tmx.local	1
1764	Jun 17, 2010 9:21:32 PM	0	0	0	1 SourceScanner	masuya	dell-83800-22.tmx.local	1
1763	Jun 17, 2010 9:20:46 PM	0	1	0	0 SourceScanner	masuya	dell-83800-22.tmx.local	1
1762	Jun 17, 2010 9:04:55 PM	0	0	1	1 SourceScanner	masuya	dell-83800-22.tmx.local	1
1761	Jun 17, 2010 8:52:56 PM	0	0	1	1 SourceScanner	masuya	dell-83800-22.tmx.local	1
1760	Jun 17, 2010 8:22:35 PM	0	7	8	C++test	masuya	dell-83800-22	15
1759	Jun 17, 2010 8:12:25 PM	0	5	1	C++test	masuya	dell-83800-22	6
1758	Jun 17, 2010 7:37:51 PM	0	2	13	C++test	masuya	dell-83800-22	15
1757	Jun 17, 2010 7:37:06 PM	0	8	7	C++test	masuya	dell-83800-22	15
1756	Jun 16, 2010 8:37:57 PM	0	2	5	C++test	masuya	dell-83800-22	7
1755	Jun 16, 2010 8:31:38 PM	0	7	8	C++test	masuya	dell-83800-22	15
1754	Jun 16, 2010 8:28:05 PM	0	7	8	C++test	masuya	dell-83800-22	15
1753	Jun 16, 2010 8:13:41 PM	0	7	8	C++test	masuya	dell-83800-22	15
1752	Jun 16, 2010 7:38:36 PM	0	0	1	1 SourceScanner	masuya	dell-83800-22.tmx.local	1
1503	May 20, 2010 1:57:53 PM	0	1	0	C++test	nagata	dell-m6300-1	1
1257	May 17, 2010 3:53:41 PM	0	0	1	1 SourceScanner	nagata	dell-m6300-1.tmx.local	1
1256	May 11, 2010 6:10:04 PM	0	8	0	C++test	kylie	kylie-83c7644dc	8
1255	May 11, 2010 6:09:23 PM	0	2	4	C++test	kylie	kylie-83c7644dc	6

この画面を表示するには: テスト > 最近のテスト ログ

[最近のテスト ログ] レポートの表には、次の情報が表示されます。

- **ログ ID** 表示されているログに割り当てられた識別番号。クリックすると、[ログの詳細] レポートが表示されます。
- **日付** 表示されているログの日付。
- **未完了のテスト** 対応するログにおける未完了のテストの数。
- **失敗したテスト** 対応するログにおける失敗したテストの数。
- **成功したテスト** 対応するログにおける成功したテストの数。
- **ツール** Report Center データベースにログを送ったツールの名前。
- **ユーザー** ログを送信したユーザーの名前。

- **マシン** ログを送信したマシンの名前。
- **テスト合計** 対応するログで実行されたテストの総数。

列ヘッダーをクリックして、[最近のテスト ログ] レポートの表のデータをソートすることができます。

[ログの詳細] レポート

[ログの詳細] レポートには、ID、ツール、ホスト、開始日と終了日といったログ情報とこのログに含まれるテスト グループが表示されます。実行されたグループごとに、以下の情報が表示されます。

- テスト グループ ユーザー属性とテスト結果
- 失敗したテスト、未完了のテスト、成功したテストのそれぞれの数、実行されたテストの総数
- カバレッジ率 (単体テスト結果にカバレッジが表示される場合)

図 35: [ログの詳細] レポート

ログの詳細					
概要情報:					
ログ ID:	1256				
ツール:	C++test v. 7.3.1.61				
ユーザー:	kylie				
ホスト:	kylie-83c7644dc				
プラットフォーム:	Windows XP, 5.1 (x86)				
開始日:	May 11, 2010 6:10:04 PM				
停止日:	May 11, 2010 6:10:23 PM				
テスト グループ ユーザー属性 (キー, 値):					
	テスト グループ名: Coding Standards - ATM0507 (1254)				
Project	ATM_demo				
結果:					
テスト グループ名	失敗したテスト	未完了のテスト	成功したテスト	テスト合計	%カバレッジ
Coding Standards - ATM0507 (1254)	8	0	0	8	100% [259/259]

この画面を表示するには: [テスト](#) > [最近のテスト ログ](#) > [ログ ID]

[ログの詳細] レポートには次の情報が表示されます。

- **ログ ID** ログに割り当てられた識別番号。
- **ツール** Report Center データベースにログを送信したツールの名前。
- **ユーザー** ログを送信したユーザーの名前。
- **ホスト** ログを送信したマシンの名前。
- **プラットフォーム** ログが作成されたプラットフォームの名前。
- **開始日** このログのテストが開始した日。

- **停止日** このログのテストが終了した日。
- **テスト グループ ユーザー属性 (キー , 値)** ログに含まれるすべてのテスト グループのキーと値。
- **テスト グループ名** テスト グループに割り当てられた名前。クリックすると、[テスト グループの詳細] ページが表示されます。詳細については 「[テスト グループの詳細] レポート」を参照してください。
- **失敗したテスト** グループ内で失敗したテストの数。
- **未完了のテスト** グループ内で未完了のテストの数。
- **成功したテスト** グループ内で成功したテストの数。
- **テスト合計** グループ内のテストの総数。
- **% カバレッジ** 単体テストの行カバレッジ率 (適応可能な場合)。
注意! カバレッジは単体テストに適用されます。

[テスト グループの詳細] レポート

[テスト グループの詳細] レポート (図 36) には、テスト グループの詳細が表示されるほか、テスト グループが実行されたツール、ホスト、およびプラットフォームについての情報が表示されます。また、グループに含まれる各テストのリストも表示されます。

図 36: [テスト グループの詳細] ページ

テスト グループの詳細

概要情報

テスト グループ ID: 1254 テスト グループ名: Coding Standards - ATM0507 ログ ID: 1256 開始日: 2010-05-11 18:10:05 停止日: 2010-05-11 18:10:22	ツール: C++test ユーザー: kylie ホスト: kylie-83c7644dc プラットフォーム: Windows XP
---	---

[任意](#)
[\[未完了 \]](#)
[\[失敗 \]](#)
[\[成功 \]](#)

テスト概要:

テスト合計	未完了のテスト	失敗したテスト	成功したテスト	%カバレッジ
8	0	8	0	-

テスト

テスト ID:	名前	ステータス	未完了のルール	失敗したルール	成功したルール	%カバレッジ
1271	/ATM0507/ATM.cxx	FAILED	0	10	253	-
1284	/ATM0507/ATM.hxx	FAILED	0	5	258	-
1270	/ATM0507/Account.cxx	FAILED	0	3	260	-
1266	/ATM0507/Account.hxx	FAILED	0	6	257	-
1289	/ATM0507/Bank.cxx	FAILED	0	6	257	-
1285	/ATM0507/Bank.hxx	FAILED	0	3	260	-
1268	/ATM0507/BaseDisplay.cxx	FAILED	0	4	259	-
1267	/ATM0507/BaseDisplay.hxx	FAILED	0	2	261	-

この画面を表示するには: テスト > 最近のテスト ログ > [日付] > [テスト グループ名]
 [テスト グループの詳細] ページには次の情報が表示されます。

- **テスト グループ ID** テスト グループに割り当てられた識別番号。
- **テスト グループ名** テスト グループに割り当てられた名前。
- **ログ ID** ログに割り当てられた識別番号。
- **開始日** このグループに含まれるテストの実行が開始した日。
- **停止日** このグループに含まれるテストの実行が終了した日。
- **ツール** Report Center データベースにログを送信したツールの名前。
- **ユーザー** ログを送信したユーザーの名前。
- **ホスト** ログを送信したホストの名前。
- **プラットフォーム** テスト グループが実行されたプラットフォーム。
- **テスト合計** グループ内に含まれるテストの総数。
- **未完了のテスト** グループ内で未完了のテストの数。
- **失敗したテスト** グループ内で失敗したテストの数。
- **成功したテスト** グループ内で成功したテストの数。

- **Total Tests** グループ内のテストの総数。
- **%Coverage** 単体テストの行カバレッジ率 (適応可能な場合)。
- **テスト ID** 表示中のグループに含まれるテストに割り当てられた識別番号。
- **名前** 表示中のグループに含まれるテストの名前。クリックすると、[テストの詳細] レポートが表示されます。詳細については 「[テストの詳細] レポート」を参照してください。
- **ステータス** 表示中のグループに含まれるテストのステータス。
- **未完了のテスト ケース** テストが生成した、未完了のテスト ケースの数。
- **失敗したテスト ケース** テストが生成した、失敗したテスト ケースの数。
- **成功したテスト ケース** テストが生成した、成功したテスト ケースの数。
- **% カバレッジ** 単体テストの行カバレッジ率 (適応可能な場合)。

[テストの詳細] レポート

[テストの詳細] レポート (図 37) には、選択されたテストについての情報と、ログに送られたすべてのメッセージの詳細なリストが表示されます。

図 37: [テストの詳細] レポート

テストの詳細:

概要情報

日付: 2010-05-11 18:10:15
 テストグループ名: Coding Standards - ATM0507
 テストID: 1269
 テスト名: ATM0507/Bank.cpp
 テストカバレッジ: -

すべてのメッセージ: 8
 メッセージ:

メッセージID	メッセージ	ステータス	エラータイプ	ファイル	行	ユーザー	%カバレッジ
2537	new 演算子を使用してはならない	FAILED	Coding Violation	Bank.cpp	35	kylie	-
2538	符号付き型と符号なし型の間の暗黙的変換を使用してはならない	FAILED	Coding Violation	Bank.cpp	18	kylie	-
2539	The basic numerical type 'int' should not be used	FAILED	Coding Violation	Bank.cpp	15	kylie	-
2540	パラメータ num を const として宣言する	FAILED	Coding Violation	Bank.cpp	15	kylie	-
2541	パラメータ password を const として宣言する	FAILED	Coding Violation	Bank.cpp	15	kylie	-
2542	Account 型への C スタイルのキャストが使用されている	FAILED	Coding Violation	Bank.cpp	20	kylie	-
2543	ローカル変数 user Account を const として宣言する	FAILED	Coding Violation	Bank.cpp	35	kylie	-
2544	1 つの式の中で ++ および -- 演算子を他の演算子とともに使用してはならない	FAILED	Coding Violation	Bank.cpp	36	kylie	-

2537 FAILED new 演算子を使用してはならない

属性 **値**

status failure
 error_type Coding Violation
 severity 8
 line 85
 test_category MISRA C++ 2008
 rule_id MISRA2008-18_4_1
 description 動的なヒープ メモリの割り当てを使用してはならない
 message new 演算子を使用してはならない
 ownership_line kylie
 file C:\Documents and Settings\kylie\Parasoft\Wcpptest\workspace\ATM0507\Bank.cpp
 symbol C:\Documents and Settings\kylie\Parasoft\Wcpptest\workspace\ATM0507\Bank.cpp

2538 FAILED 符号付き型と符号なし型の間の暗黙的変換を使用してはならない

属性 **値**

status failure
 error_type Coding Violation
 severity 8

この画面を表示するには: テスト > 最近のテスト ログ > [日付] > [テストグループ名] > [名前] (テスト名)

[テストの詳細] レポートには次の情報が表示されます。

- **日付** 表示中のテストが実行された日。
- **テストグループ名** 表示中のテストが属するグループに割り当てられた名前。
- **テストID** 表示中のテストに割り当てられた識別番号。
- **テスト名** 表示中のテストに割り当てられた名前。
- **テストカバレッジ** 表示中のテスト (ファイル) の行カバレッジ。
- **すべてのメッセージ** 表示中のテストの実行によって生成されたメッセージの数。
- **メッセージID** メッセージに割り当てられた識別番号。
- **メッセージ** 表示中のテストの実行によって生成されたサマリーテキスト。クリックすると、メッセージについての詳細 (属性とその値のリスト) に移動します。

- **ステータス** メッセージのステータス (成功、失敗、未完了)。
- **エラー タイプ** テストによって生成されたエラーのタイプ。エラーが生成されなかった場合、「No Error」と表示されます。
- **ファイル** メッセージが格納されるファイルの名前。
- **ユーザー** 表示中のテストを実行したユーザーの名前。
- **% カバレッジ** 表単体テストの行カバレッジ率 (適用可能な場合)

変更ベースのテスト レポート

変更ベースのテスト レポートは、最近のソース コードの変更に応じて、何を再テストするべきかをレポートします。

変更ベースのテストは、ソース コードの変更によって影響を受けるテスト、要求、タスク、および欠陥を正確に特定します。コードを変更するたびにシステム全体を再テストする必要がないため、生産性が非常に向上します。

以下の方法で再テストが必要なテストを特定できます。

- 相互に関係付けられたコードの変更によって、どの手動テスト シナリオを再テストするべきかを確認するには、[テスト] > [変更ベースのテスト] > [テスト シナリオ] を選択します。
- 相互に関係付けられたコードの変更によって、どの要求 / 欠陥を再テストするべきかを確認するには、[テスト] > [変更ベースのテスト] > [要求 / 欠陥] を選択します。

テストはどのように推奨されるのか

Development Testing Platform はプロジェクト機能にどのソース コードが関連するかをトラッキングします。ソース コードに変更があると、変更によって不具合が発生していないかどうかをチェックするために、関連する実装を再テストするべきであることを Development Testing Platform は分かります。

開発者が要求を実装し、欠陥を修正する間、Development Testing Platform は関連するソース コードをモニタリングし、対応する要求 / 欠陥にこのソース コードを関連付けます。その後、この実装または修正された機能は通常 QA によってテストされます。

多くの場合、すでに開発してテストした作業に関連するコード部分を後で修正することはよくあります。この変更によって、すでに検証された機能が変化したり不具合が発生する場合があります。

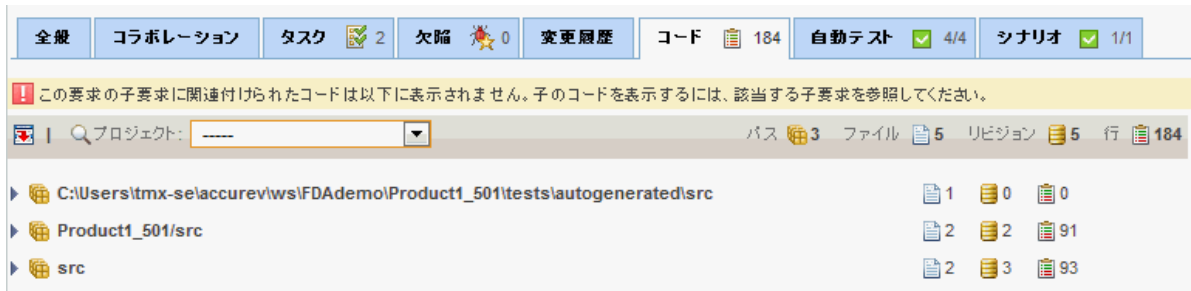
推奨されるテスト レポートを使用すると、既存の機能がコードの変更後に正しく動作することを確認するために、どの機能の再テストが必要かを判断するのに役立ちます。プロジェクト成果物（要求、欠陥、改善）に関連するソース コードが特定の期間に変更されると、その成果物は「再テストが必要」としてマークされます。再テストのタスクには優先順位が付けられます。ソース コードの変更が多い項目に、最も高い優先順位が割り当てられます。

ソース コードはどのようにトラッキングされるのか

Development Testing Platform は、特定の機能を実装するときに開発または変更されたソース コードをトラッキングします。コードは次の 2 種類の方法でトラッキングすることができます。

- ソース管理システムにチェックインするときに、特別な記号 @task taskID を開発者が追加する（詳細については「タスクに対する作業」を参照）。
- Development Testing Platform タスク アシスタント (Parasoft Test で利用可能) を介して開発者がタスクに対して作業する。

特定の成果物（要求、欠陥、改善）の詳細ページの [コード] タブに、コードの相関関係が表示されます。



トラッキングされたソース コードに変更があると、Development Testing Platform はどの機能（要求、欠陥、手動テスト）を再テストするべきかを判断し、[推奨されるテスト] レポートでユーザーに通知します。

[推奨されるテスト] レポートには、2つの観点から再テストすべき機能が表示されます。

- テストごとの推奨されるテスト
- 欠陥 / 改善ごとの推奨されるテスト

変更ベースのテスト：テスト シナリオ

463 ページの「Project Center でのテスト」で説明している手動テストを前提として説明します。

[変更ベースのテスト：テスト シナリオ] レポートには、以前にテストした機能に関するソースコードが変更されたため、再テストするべきである手動テスト シナリオが表示されます。言い換えると、手動テスト シナリオ A が要求 B に関係付けられていて、要求 B に関連するソースコードに変更があった場合、手動テスト シナリオ A は「再テストが必要」としてマークされます。

欠陥

	ID	作成日	概要	優先度	重要度	割り当て:	ステータス	Report Center ステータス	テストが必要	
1	54	2012-08-09	characters	Medium	Medium		New	未解決	はい	
2	50	2012-08-02	Forgot password reactivation email functionality does not work	High	High	JohnSmith	Assigned	未解決	はい	
3	51	2012-08-02	National characters do not work in user profile data	Medium	Medium	DavidScott	Resolved	修正	はい	111
4	46	2012-06-12	Forgot password reactivation email functionality does not work	High	High	JohnSmith	Assigned	未解決	はい	
5	47	2012-06-12	National characters do not work in user profile data	Medium	Medium	DavidScott	Resolved	修正	はい	61

[優先度] の列は、優先度によってテストをソートします。テストに関連する機能への変更が多ければ多いほど、その優先度が高くなります。

[リレーション] の列には、シナリオが関係付けられているプロジェクト成果物が表示されます。

レポートの対象期間を指定できます。[最新 Last 7 | 15 | 30 日] および [最新 Last 12 | 26 | 52 週] の数字をクリックするか、[範囲モードに切り替え] アイコンをクリックして期間の範囲を指定します。

優先度のセルの値をクリックすると、コードについての詳細が表示されます。

例：

Scenario: Test recording alerts in Chrome
Version: 0
Scenario Recent Run: Apr 15, 2011

The following file revisions have been committed to repository since the scenario recent run

File	Version	Revision Date	Relation
HttpUnitContext.java	1.24	Apr 19, 2011	REQ 1047
FFBrowserController.java	1.146	Apr 15, 2011	REQ 1047
AjaxTestingContext.java	1.59	Apr 19, 2011	REQ 1047
ProxyEventHandler.java	1.35	Apr 15, 2011	TASK 34696
ProxyEventHandler.java	1.35	Apr 15, 2011	REQ 1047
Connection.java	1.61	Apr 19, 2011	TASK 34696

詳細レポートには、以下のすべてのソース コード リビジョンが表示されます。

- 特定の機能に関連するリビジョン
- 最近の手動テストの実行の後にソース管理にコミットされたリビジョン

変更ベースのテスト：要求 / 欠陥

[変更ベースのテスト：要求 / 欠陥] レポートには、関係するソース コードが変更されたため、再テストすべきである要求 / 欠陥が表示されます。言い換えると、要求 A に関係付けられているコードに変更があった場合、要求 A は「再テストが必要」としてマークされます。

The following Requirement(s)/Defect(s) should be retested:

Id	Summary	Priority ▲	#Revisions	Last modification
REQ 1047	Support Chrome	Very high	34	Apr 19, 2011
PR 95250	navigate test failing when using proxy that requires basic authentication	Very high	6	Apr 19, 2011
PR 94760 task 35606	- Make sure finish() is called each time an ErrorOutputBatchContext is created, to prevent memory leaks	High	3	Apr 15, 2011
PR 74683 task 35823	- Cardinal Health/ExactTarget/Verisign/EMC/Fidelity/Lockheed/CIBC - Environment Variables do not work for File Paths	Medium	2	Apr 15, 2011
PR 85984 task 35826	- Sabre - Allow Environment Variables in Delay fields for tests	Low	1	Apr 15, 2011
PR 93892 task 35627	- CCS - Unable to navigate through HTML report using japanese characters in IE	Low	1	Apr 20, 2011

優先度のセルの値をクリックすると、変更があったエンティティについての詳細が表示されます。

例：

Entity Id: PR 94760
Summary: task 35606 - Make sure finish() is called each time an ErrorOutputBatchContext is created, to prevent memory leaks
Total Revisions: 4

The following file revisions have been committed to repository during last 7 days

File ▲	Revision	Last modification
BatchContextUtil.java	32.2	Apr 15, 2011
TestCleanupTest.java	32.2	Apr 15, 2011
ToolTest.java	32.308	Apr 15, 2011
ToolTest.java	32.309	Apr 15, 2011

詳細レポートには、以下のすべてのソース コード リビジョンが表示されます。

- 特定の要求 / 欠陥に関連するリビジョン
- 特定の期間にソース管理にコミットされたリビジョン

コードメトリクスレポート

Development Testing Platform (DTP) には、Cyclomatic Complexity、クラスの数、メソッドの数、静的属性の数といった、ソースコードメトリクスデータ（統計）を表示するレポートがあります。これらのメトリクスはコード解析ツールによって計測され、DTP に送信されます。

このメトリクス情報から、各コードレベル（パッケージ、クラス、メソッド）についての詳細を参照したり、メトリクスのドキュメントを表示したり、メトリクスが適用されるソースコードを参照したりすることができます。

次の 2 種類のコードメトリクスレポートがあります。

- [上位のメトリクス結果] レポート
- [単一メトリクスの概要] レポート (9.x)
- メトリクスコンプライアンスレポート
- [メトリクスの概要] レポート
- [単一メトリクスの概要] レポート

管理者への注意事項 メトリクスの結果を DTP に送信する方法については『Development Testing Platform 管理者マニュアル』を参照してください。

[上位のメトリクス結果] レポート

DTP は最も高いメトリクス値を持つプロジェクトのパッケージ、クラス、メソッドを記録し、[上位のメトリクス結果] レポートにリストします。プロジェクトでの許容レベルを値が超える場合、そのようなパッケージ、クラス、メソッドは調査の対象となるべきです。これは、その部分のコードのリファクタリングが必要なことを表している場合があります。

デフォルトでは、レベルごとに 20 個の項目が表示されます。この設定は必要に応じて変更できません。

図 38: [上位のメトリクス結果] レポート - パッケージ

Metric Top Results				
Project:	<input type="text" value="Peru"/>			
Selected Metric:	<input type="text" value="Cyclomatic Complexity"/>	[Show Metric Description]		
Drop Date:	2007-01-18			
Results ordered by Mean (descending).				
Package Level:				
Package Name	Sum	Mean	Number Of Items	Standard Deviation
 com.parasoft.grs.processors.xreportprocessors.tcm.suppressions	-	6.333	6	4.57
 com.parasoft.grs.processors.storage.sourcecode	-	5.905	21	6.233
 com.parasoft.grs.processors.storage.manualtests	-	5.857	42	3.392
 com.parasoft.grs.processors.storage.metrics	-	5.75	4	4.969
 com.parasoft.grs.processors.storage.ls	-	5.286	7	4.978
 com.parasoft.grs.processors.xreportprocessors.managerdashboard	-	5.136	88	6.582

図 39: [上位のメトリクス結果] レポート - クラス








Class Level:				
Class Name	Sum	Mean	Number Of Items	Standard Deviation
 TestDetailsStorage	-	20.667	3	23.697
 TestCaseInfo	-	12	2	11
 SuppressionSaver	-	10.333	9	22.998
 Storage	-	9.75	4	10.686
 CodingStandardDetailsStorage	-	9.286	7	7.814
 ErrorsByFileDeveloperStorage	-	9.125	8	8.536
 ManualTestsOverviewProcessor	-	8.545	11	7.844

図 40: [上位のメトリクス結果] レポート - メソッド

Method Level:				
Method Signature	Sum	Mean	Number Of Items	Standard Deviation
 saveSuppression(long, java.sql.Timestamp, int, int, int, int, int, int, int, int, long, int, java.lang.String)	75	75	1	0
 getDetails(com.parasoft.grs.processors.xreportprocessors.TestDetails\$PropertiesWrapper, com.parasoft.grs.processors.xreportprocessors.TestDetails\$TestSummaryInfo)	54	54	1	0
 prepareDataElement(com.parasoft.grs.rserver.xreportengine.XReportDocument, java.util.Map, com.parasoft.grs.rserver.xreportengine.XReportRuntimeProperties)	48	48	1	0
 createMaxUsageData(com.parasoft.grs.rserver.xreportengine.XReportDocument,				

この画面を表示するには : 監査 > 上位のメトリクス結果

[上位のメトリクス結果] レポートには次の情報が表示されます。

- プロジェクト Parasoft ツールによって定義され、送信されたプロジェクトの名前。
- 選択されたメトリクス 値を表示するメトリクス。

[メトリクスの詳細を表示] クリックすると、現在選択しているメトリクスについての説明がブラウザーに表示されます。

- **ドロップ日** 最後のメトリクス ドロップの日付。言い換えると、選択されたプロジェクトに対してメトリクス解析が実行された最後の日。
- **名前 / シグニチャ** テスト対象のパッケージ / クラスの名前、またはテスト対象メソッドのシグニチャ。
- **合計** テスト対象オブジェクトの選択されたメトリクス値の合計。多くのメトリクスでこのフィールドは空白です。
- **平均** 選択されたメトリクスのテスト対象オブジェクトの算術平均。
- **アイテムの数** 特定のテスト対象オブジェクトに属する子ノードの数。
- **標準偏差** 特定のメトリクス結果の統計的標準偏差。

レポート中の項目をクリックすると、単一メトリクスのレポートが開き、さらに詳しい情報が表示されます。110 ページの「[単一メトリクスの概要] レポート (9.x)」を参照してください。

[単一メトリクスの概要] レポート (9.x)

[単一メトリクスの概要] レポートには、Parasoft 9.x ツールから収集された、特定のプロジェクトに関連するデータとドロップダウン リストから選択されたメトリクスが示されます。このレポートには DTP Engine からのデータは含まれません。

1. レポート ビューで [監査] メニューの [単一メトリクスの概要] をクリックすると、[単一メトリクスの概要] レポートが開きます。
2. ドロップ ダウン メニューからプロジェクトとメトリクスを選択します。

テスト ユニット (パッケージ、クラス、メソッド) がツリー形式で表示され、ユーザーはプロジェクト パッケージから 1 つのメソッドまで階層ツリーをたどることができます。テスト ユニットごとに次の 4 種類のメトリクス値が示されます。

- 合計
- 平均
- 項目の数
- 標準偏差

これらの値は、表中でテスト対象オブジェクトごとにその隣に表示されます。

図 41: [単一メトリクスの概要] レポート

Single Metric Overview				
Path:	[Peru] / com.parasoft.grs.common /			
Project:	Peru			
Selected Metric:	Cyclomatic Complexity [Show Metric Description]			
Current Drop:	2007-01-25			
Drop To Compare:	2007-01-18			
Test Unit:				
Name	Sum	Mean	Number Of Items	Standard Deviation
com.parasoft.grs.common	- (-)	2.359 (+0)	1,336 (+0)	2.835 (+0)
Subpackages:				
Name	Sum	Mean	Number Of Items	Standard Deviation
com	- (-)	- (-)	- (-)	- (-)

[単一メトリクスの概要] レポートには次の情報が表示されます。

- **パス** ソース階層中の現在の位置 (階層中のレベルとノード)
- **プロジェクト** Parasoft ツールによって定義されて送信されたプロジェクトの名前。
- **選択されたメトリクス** 値を表示するメトリクス名。

[**メトリクスの詳細を表示**] クリックすると、現在選択されているメトリクスについての情報がブラウザーに表示されます。

- **現在のドロップ** 最後のメトリクスのドロップの日付。言い換えると、選択されたプロジェクトに対してメトリクス解析が実行された最後の日。
- **比較するドロップ** 現在のドロップ値と比較するのに使用されるドロップの日付。
 - 10 を超えるドロップが利用可能な場合、10 個目の最も古いドロップが比較に使用されます。
 - 利用可能なドロップの数が 10 より少ない場合、最も古いドロップが比較に使用されます。

[単一メトリクスの概要] レポートには次の情報が含まれます。

- **テストユニット / パッケージ / クラス の表** 親テスト対象オブジェクトの結果が表示されます。子ノードの親テスト対象オブジェクトは表にリストされます (図 42)。
- **サブパッケージ (オプション) の表** パッケージ内の親テスト対象オブジェクトにサブパッケージがある場合に表示されます。サブパッケージのメトリクス値が表示され、1 つのメソッドまで情報をたどることができます。

メトリクス値には次の情報が含まれます。

- **名前** テスト対象オブジェクトの名前。
- **合計** テスト対象オブジェクトの選択されたメトリクス値の合計。
- **平均** テスト対象オブジェクトの選択されたメトリクス値の算術平均。


- **アイテムの数** 特定のテスト対象オブジェクトに属する子ノードの数。
- **標準偏差** 特定のメトリクス結果の統計的標準偏差。

注意！ 括弧内の数字は「最後のドロップ」と「比較対象のドロップ」の差を表します。

図 42: ソースコードの参照

メソッド:

シグニチャ ▲	合計	平均	アイテムの数	標準偏差
 apply(Transaction)	4.000 (-)	4.000 (-)	1 (-)	0.000 (-)
 apply0(ChangeOwnership)	1.000 (-)	1.000 (-)	1 (-)	0.000 (-)
 apply0(Deposit)	1.000 (-)	1.000 (-)	1 (-)	0.000 (-)
 apply0(Withdraw)	1.000 (-)	1.000 (-)	1 (-)	0.000 (-)
 BankAccount(int, int)	1.000 (-)	1.000 (-)	1 (-)	0.000 (-)
 getBalance()	1.000 (-)	1.000 (-)	1 (-)	0.000 (-)
 getCustomerId()	1.000 (-)	1.000 (-)	1 (-)	0.000 (-)

クラスとメソッドの隣に表示される拡大鏡のアイコン  をクリックすると、そのクラス/メソッドのソースコードを参照することができます。

注意！ SourceScanner がその特定のプロジェクトソースに対して実行された後に、拡大鏡のアイコンが表示されます。

[メトリクスの概要] レポート

[メトリクスの概要] ウィジェットで [more] リンクをクリックすると、[メトリクスの概要] レポートが開きます (74 ページの「メトリクス ウィジェット」を参照)。このレポートは、[メトリクスの概要] ウィジェットで選択されたプロジェクトのメトリクスデータを表示します。

Metrics Overview

Filter: SDM Platform

General

Metric	Value
Number of Files	5826
Number of Types	7119
Number of Blank Lines in File	119564
Number of Comment Lines in File	311817
Number of Physical Lines in File	949152
Number of Source Lines in File	-
Comments / Number of Logical Lines Ratio in File	2.17

Types/Functions

Metric	Average	Maximum
Cyclomatic Complexity	2.02	66
Essential Cyclomatic Complexity	1.34	57
Modified Cyclomatic Complexity	1.99	66
Strict Cyclomatic Complexity	2.14	86

Types

Metric	Average	Maximum
Coupling Between Objects	4.98	459
Inheritance Depth of Class	1.9	7
Number of Methods in Type	6.5	264
Number of Private Members in Type	3.97	130
Number of Protected Members in Type	0.49	43
Number of Public Members in Type	5.01	264

レポートには以下のメトリクスだけが表示されます。

- 全般
 - ファイルの論理行数に対するコメントの割合
 - ファイルの空白行数
 - ファイルのコメント行数
 - ファイル数
 - ファイルの物理コード行数
 - ファイルのソース行数
 - 型の数
- 型 / 関数
 - Cyclomatic Complexity
 - Essential Cyclomatic Complexity
 - Modified Cyclomatic Complexity
 - Strict Cyclomatic Complexity
- タイプ
 - オブジェクト間の結合

- クラスの継承の深さ
- 型のメソッド数
- 型の Private メンバー数
- 型の Protected メンバー数
- 型の Public メンバー数

このレポートでは、以下のアクションを実行できます。

- 列ヘッダーをクリックして表をソートします。
- 表の値をクリックして [単一メトリクスの概要] レポートを開きます。詳細については 114 ページの「[単一メトリクスの概要] レポート」を参照してください。

[単一メトリクスの概要] レポート

[単一メトリクスの概要] レポートは、選択したプロジェクトの複数のモジュールにわたって、1 つのメトリクスの集計データを表示します。集計データはヒート マップで表示され、メトリクス値が大きいほど、より大きく色の濃いタイルとしてデータが示されます。逆に、小さく色の薄いタイルはメトリクス値が小さいことを表します。このレポートは、表形式でデータを表示することもできます。このレポートを表示するには、[メトリクスの概要] ウィジェットで値をクリックするか、[トップ 10 モジュール - ツリー マップ] ウィジェットで [more] リンクをクリックします (74 ページの「メトリクス ウィジェット」を参照)。

Single Metric Overview

Filter: SDM Platform Metric: Number of Physical Lines in File

		Average	Minimum	Maximum	Sum										
com.parasoft.pst.planning.usecases	com.parasoft.grs.datacollector	com.parasoft.grs.admin	com.parasoft.concerto.codereview	com.parasoft.grs.rserver	com.parasoft.sdm.storage.managers	com.parasoft.pst.jsf	com.parasoft.sdm.api.sourcecon	com.parasoft.sdm.licensecli	com.parasoft.apidtp.processors	com.parasoft.pst.reporting.api	com.parasoft.pst.planning.cli	com.parasoft.sdm.api.defects	com.parasoft.api.dtp.processor		
	com.parasoft.pst.planning.tasks		com.parasoft.grs.rserver.sox	com.parasoft.sdm.build.number	com.parasoft.grs.common	com.parasoft.sdm.client	com.parasoft.pst.security.impl	com.parasoft.pst.security	com.parasoft.pst.sdm.r	com.parasoft.pst.sdm.j	com.parasoft.pst.sdm	com.parasoft.pst.sdm	com.parasoft.pst.sdm	com.parasoft.pst.sdm	
com.parasoft.xweb.common	com.parasoft.pst.planning.requirements	com.parasoft.pst.planning.soop	com.parasoft.grs.charts	com.parasoft.pst.planning.core	com.parasoft.pst.planning.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	
	com.parasoft.pst.planning.requirements	com.parasoft.pst.planning.soop	com.parasoft.grs.charts	com.parasoft.pst.planning.core	com.parasoft.pst.planning.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	
com.parasoft.pst.planning.bugs	com.parasoft.pst.planning.requirements	com.parasoft.pst.planning.soop	com.parasoft.grs.charts	com.parasoft.pst.planning.core	com.parasoft.pst.planning.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	
	com.parasoft.pst.planning.requirements	com.parasoft.pst.planning.soop	com.parasoft.grs.charts	com.parasoft.pst.planning.core	com.parasoft.pst.planning.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	
com.parasoft.grs.rserver.scenarios	com.parasoft.pst.planning.requirements	com.parasoft.pst.planning.soop	com.parasoft.grs.charts	com.parasoft.pst.planning.core	com.parasoft.pst.planning.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	
	com.parasoft.pst.planning.requirements	com.parasoft.pst.planning.soop	com.parasoft.grs.charts	com.parasoft.pst.planning.core	com.parasoft.pst.planning.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	com.parasoft.pst.reporting.web	

Module	Avg ▼	Min	Max	Sum	
com.parasoft.pst.planning.usecases	662.75	103	3624	7953	▲
com.parasoft.xweb.common	348	46	650	696	
com.parasoft.pst.planning.bugs	328.43	13	3269	4598	
com.parasoft.grs.rserver.scenarios	325.57	42	1308	6837	
com.parasoft.grs.datacollector	315.99	21	2439	23699	

このレポートでは、以下のアクションを実行できます。

- データについて異なる集計を表示します (メソッド レベルおよび型レベルのメトリクスのみ)。[平均]、[最小]、[最大]、[合計]をクリックすると、ヒートマップビューが変化します。集計を変更すると、選択した集計で表もソートされます。ファイルレベルのメトリクスの場合、利用できるのは [合計] だけです。
- ヒートマップでタイルをクリックするか、[モジュール]列でリンクをクリックすると、メトリクス エクスプローラービューにそのモジュールが表示されます。148 ページの「メトリクス エクスプローラー」を参照してください。
- 表で列ヘッダーをクリックしてデータをソートします (ヒートマップは変わりません)。

セキュリティ レポート

Report Center では、[セキュリティ] メニューから セキュリティ レポートを作成することができます。

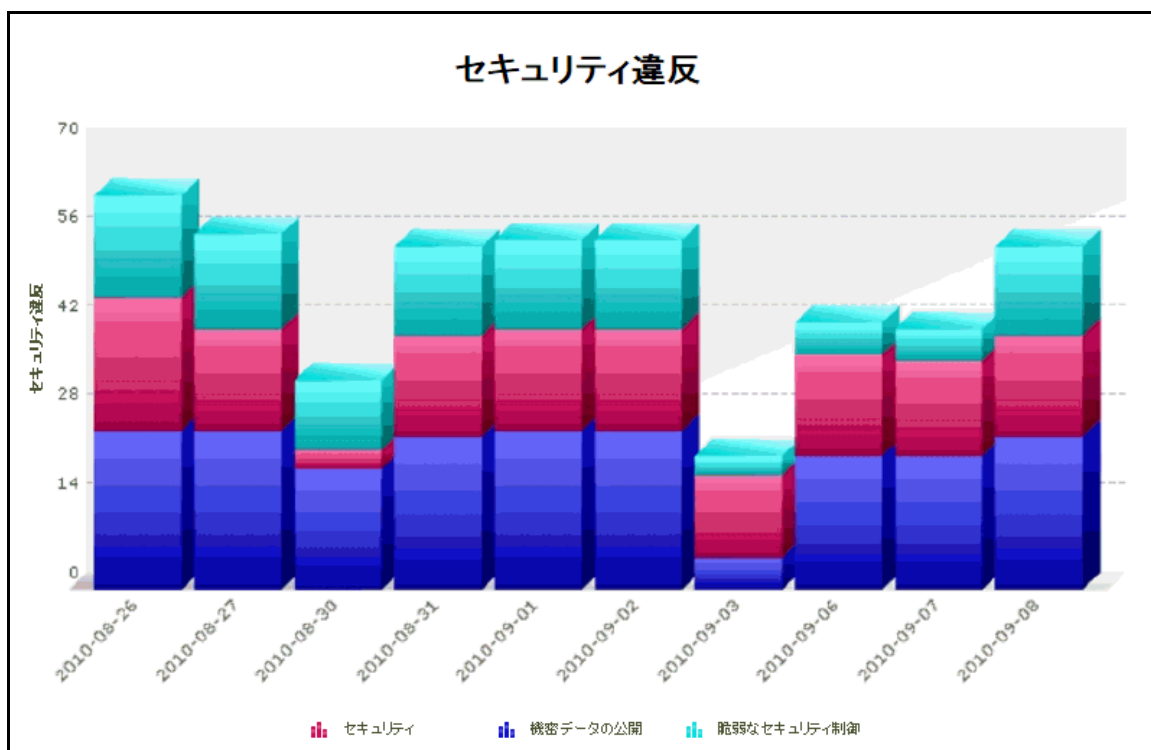
このセクションの内容：

- [セキュリティ違反] レポート
- [セキュリティ テスト] レポート

[セキュリティ違反] レポート

[セキュリティ違反] レポートを表示するには、[セキュリティ] メニューの [セキュリティ違反] をクリックします。特定の期間について [セキュリティの違反] レポートが表示されます。

図 43: [セキュリティの違反] レポートのグラフ



[セキュリティの違反] レポートには、セキュリティ タイプ エラーのさまざまなカテゴリが表示されます。グラフの棒はそれぞれ、選択されたプロジェクトについてある 1 日のテスト結果を表します。

詳細情報へのアクセス

グラフ中の棒をクリックして [セキュリティ違反] レポートの詳細を参照することもできます。クリックした日付に対する [セキュリティの詳細] レポートが表示されます。

図 44: [セキュリティの詳細] レポート

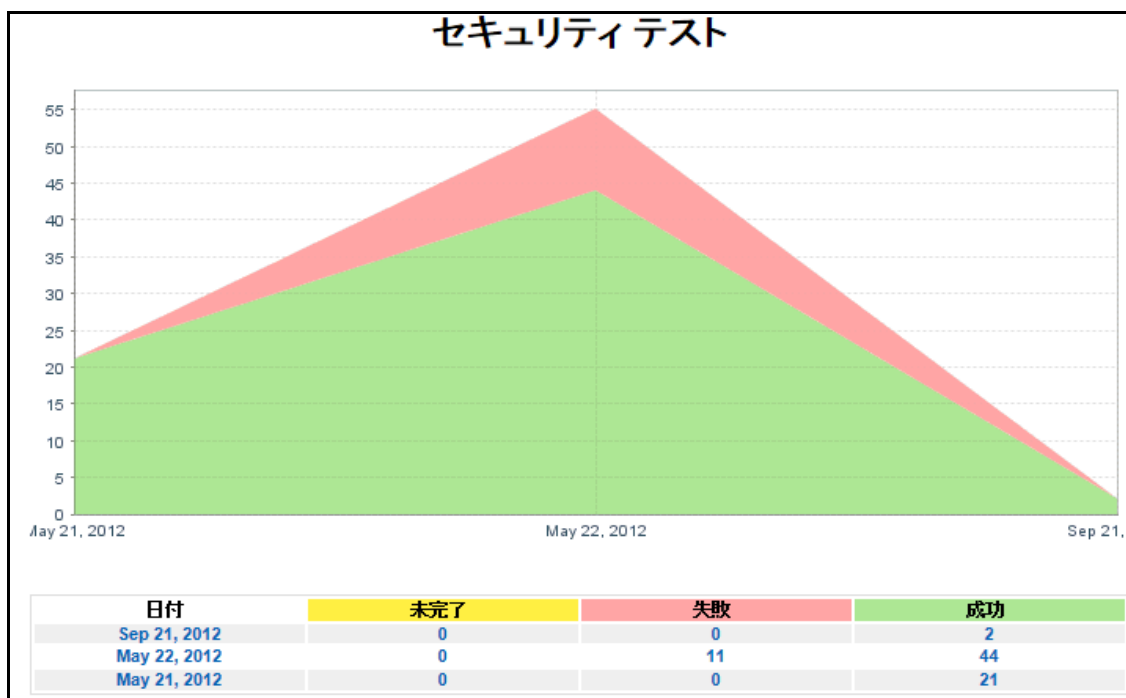


このレポートには、すべてのセキュリティ エラー カテゴリと各カテゴリに属するエラーが表示されます。ファイル名、作成者、重要度といった基本的なプロパティがエラーごとに表示されます。

[セキュリティ テスト] レポート

セキュリティ テスト レポートは、プロジェクト内で実施されたセキュリティ関連テストの情報を表示します。テスト結果や指定された期間内の傾向等を表示します。静的解析については、表示された数値は、「セキュリティ」カテゴリのコーディング スタンドールールでテストされたファイルの数を表します。

図 45:[セキュリティ テスト]レポート



[セキュリティ レポート]には、未完了、失敗、成功のセキュリティ テストの数が日付ごとに表示されます (図 45)。

詳細情報へのアクセス

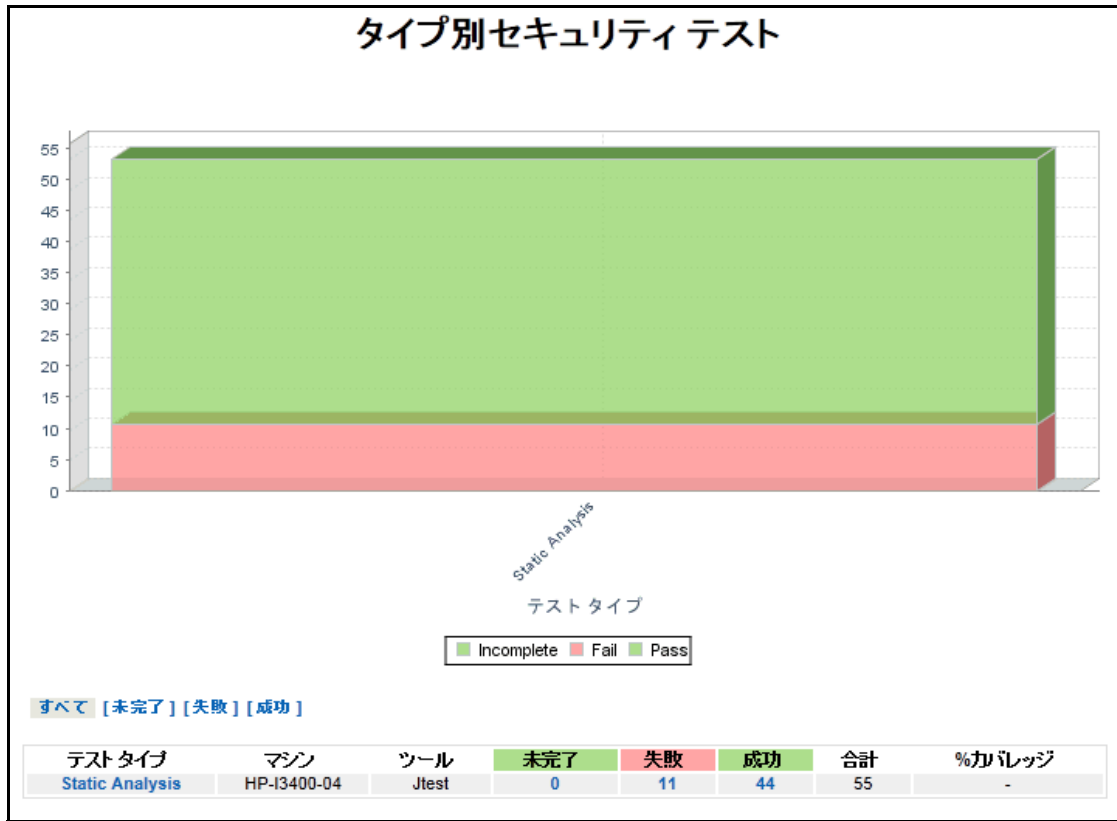
調査が必要な問題がある場合、[セキュリティテスト]レポートからさらに詳しい情報を参照できます。表の中の日付をクリックするか、または日付に対応する未完了、失敗、成功の数をクリックして[タイプ別セキュリティ テスト]レポートを表示します (図 46)。

[タイプ別セキュリティ テスト]レポート

[タイプ別セキュリティ テスト]レポートを表示するには、[セキュリティ]メニューの[セキュリティ テスト]をクリックして[セキュリティ テスト]レポートを表示し、表の中のリンクをクリックします。

[タイプ別セキュリティ テスト]レポートには、選択した日付に実行されたテストがタイプごとに表示されます。さらに、テストが実行されたマシン名とツールも表示されます。

図 46:[タイプ別セキュリティ テスト] レポート: 成功



[タイプ別セキュリティ テストの詳細] レポート

テスト、ルール、テスト ケースの統計について、さらに詳しい情報を得ることができます。コードに対して実行されたテスト グループ情報に移動し、失敗したテスト ケースのエラーを含むコード行までたどることができます。

図 47:[タイプ別セキュリティ テストの詳細] レポート

タイプ別セキュリティ テストの詳細

グループ名	マシン	ツール	未完了	失敗	成功
Coding Standards - TaskManagerService3	HP-I3400-04	Jtest	0	0	1
Coding Standards - TaskManagerService3	HP-I3400-04	Jtest	0	0	1

セキュリティ テスト グループの詳細

[タイプ別セキュリティ テストの詳細] レポートでグループ名のリンクをクリックすると、[セキュリティ テスト グループの詳細] レポートが開き、個々のテスト実行が表示されます。

図 48: [セキュリティ テスト グループの詳細] レポート

セキュリティテストグループの詳細

概要情報:

テストグループ ID:	1504	ツール:	Jtest
テストグループ名:	Coding Standards - TaskManagerService3	ユーザー:	nagata
ログ ID:	1503	ホスト:	HP-I3400-04
開始日:	2012-09-21 15:40:01	プラットフォーム:	Windows 7
停止日:	2012-09-21 15:40:05		

すべて [未完了] [失敗] [成功]

テスト概要:

テスト合計	未完了のテスト	失敗したテスト	成功したテスト	%カバーレッジ
1	0	0	1	-

テスト:

テスト ID:	名前	ステータス	未完了のメッセージ	失敗メッセージ	成功したメッセージ	%カバーレッジ
1504	/TaskManagerService3/src/jp/co/tmx/Command_Task.java	PASSED	0	0	1	-

ポリシー レポート

このレポートは、一定期間の開発ポリシーへのコンプライアンスを表示します。

1. レポート ビューを開きます。
2. [レポート]メニューの[ポリシー レポート]をクリックします。

Development Testing Platform は、ポリシーのコンプライアンス状況を自動的にモニターし、スコアを生成します。チェック対象のポリシーはカスタマイズでき、テスト結果や実装中のフィーチャー、コード サイズ、プロジェクト スケジュール、予算、コード レビュー プロセス、ビルドのモニターなどを含めることができます。プロジェクト ポリシーの構成については 154 ページの「Policy Center (レガシー) ユーザーズ ガイド」を参照してください。

プロジェクト ポートフォリオ

プロジェクト ポートフォリオを見ると、マネージャーは自分が監督するプロジェクトのステータスをすばやく把握することができます。プロジェクト ポートフォリオは次のマネージャーの基本的な質問に回答します。

- プロジェクトは予定どおり完了するか？
- チームはどの程度効率的に作業しているか？
- 自動テストでどの程度ソース コードがカバーされているか？

プロジェクトの概要

プロジェクト ポートフォリオの表 (図 49) には、ログインしているマネージャーが監督する、進行中のすべてのプロジェクトがリストされます。プロジェクト名をクリックすると、プロジェクト詳細に移動します。詳細については 123 ページの「プロジェクトの詳細」を参照してください。背景色が赤いプロジェクトには、遅延や潜在的な欠陥などの問題があり、さらに詳しい調査が必要です。

図 49: プロジェクト ポートフォリオの表

プロジェクト	遅延
Empty Project	0d/
FDAdemo	13d/3
MyProj1	168d/-2

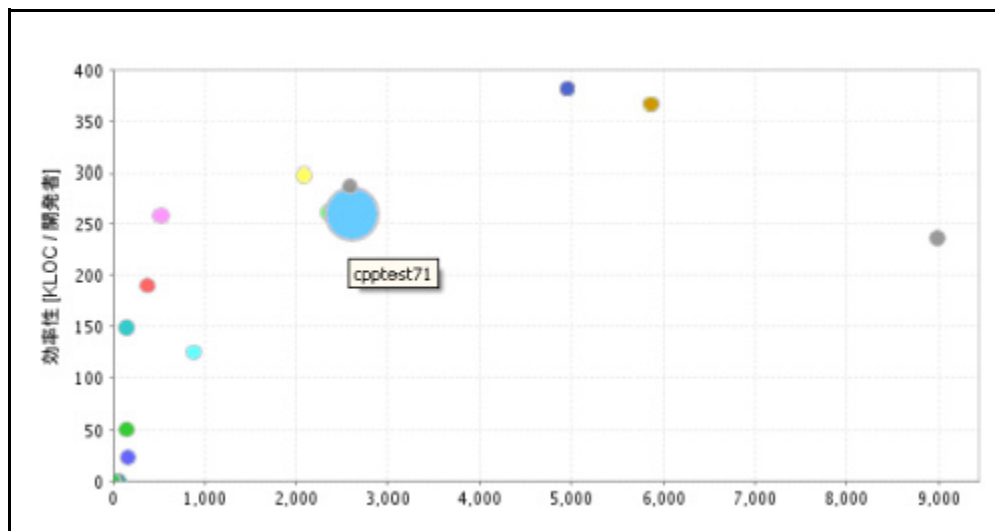
- [遅延] 列には**予定作業時間** および**予定からの乖離**が表示されます。
- 予定からの乖離が正の数の場合、プロジェクトはすでに遅延しています。赤の背景色で示されます。
- 予定からの乖離が負の数の場合、プロジェクトは予定どおり進捗しています。緑の背景色で示されます。

プロジェクトの 予定作業時間 / 予定からの乖離 をクリックすると、イテレーションのステータス レポートが表示されます。

効率性グラフ

効率性グラフ (図 50) には、特定のプロジェクトの特定の期間における作業効率が表示されます。作業効率は、プロジェクトコード行の総数と、プロジェクトに関わるチームメンバーの数に基づいて計算されます。

図 50: 効率性グラフ



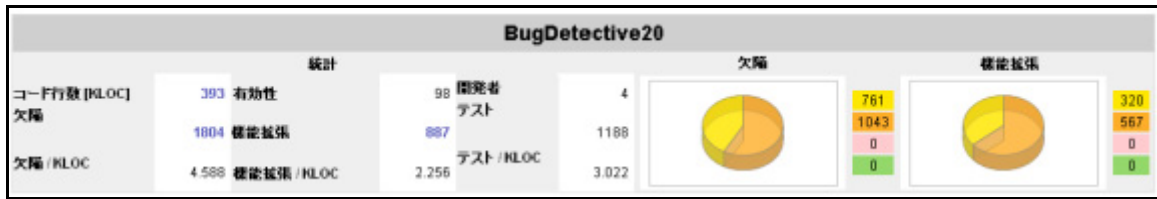
効率性グラフには、チームメンバーの作業効率に関連するプロジェクトコード行の総数が表示されます。グラフ上の丸はそれぞれ1つのプロジェクトを表します。丸がグラフの上の方にあるほど、チームメンバーは効率的に作業しています。また、丸が大きければ大きいほど、テストでコード行がカバーされています。

図 50 に表示されている C++test7.1 のプロジェクトは理想的な状態です。なぜなら、丸がグラフ中の上の方にあることから、チームメンバーは効率的に作業しています。また、丸のサイズが大きいため、チームメンバーはソースコードをカバーするために定期的にテストを作成しています。

プロジェクトの詳細

プロジェクトの詳細には、選択したプロジェクトの統計が表示されます。この表では、以下の操作を行うことができます。

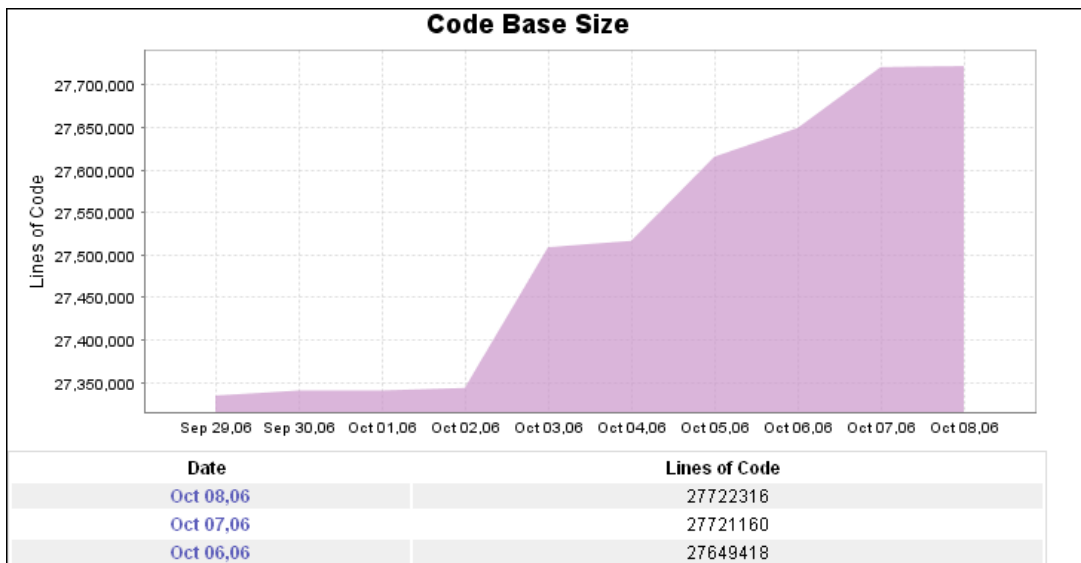
- 表のコード行数 (KLOC) セルをクリックすると、[コードベースのサイズ] レポートが開きます。詳細については 125 ページの「コードベースのサイズ」を参照してください。
- 欠陥のグラフまたはテーブルの [欠陥] セルをクリックすると、[Defects Detailed] レポートが開きます。詳細については 144 ページの「欠陥および改善レポート」を参照してください。
- 改善グラフまたはテーブルの [改善] セルをクリックすると、[Daily Enhancements] レポートが開きます。詳細については 144 ページの「欠陥および改善レポート」を参照してください。



コードベースのサイズ

コードベースのサイズレポートを表示するには、Report Center ダッシュボード ビューのコードベースのサイズ ウィジェットをクリックします。詳細については 9 ページの「コード ウィジェット」を参照してください。

[コードベースのサイズ] グラフは、コードベース中のコードの量を示します。グラフ線が上昇している場合、プログラマが作業していてコードベースにコードを追加していることを意味します。[コードベースのサイズ] グラフを見ると、日次ベースまたは週ベースでコードがチェックインされているかどうかを知ることができます。また、各週に変更されたコード、ファイル、行の量も見ることができます。実際に個々の変更を見て、ファイル中のまさに何が変更されたのかを把握できます。



[コードベースのサイズ] ページには次の情報が表示されます。

- **日付** コードがソース管理に追加された日。
[コードベースのサイズ] の表中をクリックすると、ソースコードの統計値レポートを参照できます。
- **コード行** コード行の総数。
- **[プロジェクトソースコードを表示]** このリンクをクリックすると、[プロジェクトのソースコード] レポートが表示されます。

[コードベースのサイズ] グラフをクリックすると、[ソースコードのチェックイン] グラフが表示されます。詳細については 307 ページの「ソースコードのチェックイン」を参照してください。

ソースコードの統計値

[コードベースのサイズ] レポートの表の日付をクリックすると、[ソースコードの統計値] レポートを参照できます。

Source Code Statistics

Revisions by user

User	Nov 15, 2012	Total
benken	3	3
grigorb	5	5
jfulmer	1	1
modtest	16	16
Total	25	25

Lines changed: cumulative (lines added + lines removed)

User	Nov 15, 2012	Total
benken	4	4
grigorb	94	94
jfulmer	13	13
modtest	87	87
Total	198	198

Lines changed: effective (lines added - lines removed)

User	Nov 15, 2012	Total
benken	4 (+4 / -0)	4
grigorb	54 (+74 / -20)	54
jfulmer	9 (+11 / -2)	9
modtest	31 (+59 / -28)	31
Total	98	98

ユーザー別のリビジョン

[ユーザー別のリビジョン] の表には、指定の日付 / 日付範囲の間に、指定のプロジェクトに関連するソース管理されたファイルを改訂したすべてのチームメンバーがリストされます。チームメンバーごとに、各週に行われた改訂の回数が表示されるほか、表示されている週における改訂の合計回数も表示されます。

改訂の詳細を参照するには、チームメンバーのユーザー名をクリックしてください。[ソース管理システムの概要] ページが表示されます。

変更行数

変更行数レポートには、指定の日付 / 日付範囲の間に、指定のプロジェクトに関連するソース管理ファイルを改訂したすべてのチームメンバーがリストされます。チームメンバーごとに、各週に行われた改訂の回数が表示されます。2つの変更行数ビューがあります。

- **累積**：このビューの合計行数は、削除された行数の調整を行っていません。
- **実効**：このビューの合計行数は、コードの純行数を反映しています。合計行は削除された行数の調整が行われています。

コード行の変更についての詳細を参照するには、チームメンバーの名前をクリックします。[ソース管理システムの概要] ページが表示されます。

ソース管理システムの概要

[ソース コードの統計値] レポートの表でユーザーをクリックすると、[ソース管理システムの概要] ページが表示されます。ソース管理システムの概要レポートの情報は、ファイルまたはディレクトリ単位で見ることができます。ビューを切り替えるには、[ファイル] または [ディレクトリ] をクリックします。

Source Control Summary

Date: Nov 15, 2012
 User: modtest
 File Revisions: 16
 Lines Changed: 87 (+59/-28)
 Token Changed: 152 (+129/-23)

Files Directories

[ソース管理システムの概要] ページには次の情報が表示されます。

- **日付** 選択された日付。[ソース コードの統計値] のグラフで指定され、元は [コードベースのサイズ] グラフから選択された日付。
- **ユーザー** [ソース コードの統計値] のグラフで選択されたユーザーの名前。
- **ファイル リビジョン** 選択されたユーザーによって選択された日付に更新されたファイルの数。
 - **レビューアーのタスク** 選択されたユーザーについて、選択された日付 (または期間) にこのファイルに対して行われたコード レビューの回数。
 - **課題** 選択されたユーザーについて、選択された日付 (または期間) にこのファイルに追加されたコード レビューの課題の数。
 - **レビューアーのタスクによってカバーされたファイル リビジョン** 以下の式でカバレッジを計算します。

$$(' \text{レビューアーのタスク}' \text{の合計} / \text{ファイルの更新の合計}) * 100$$
 このレポートは、3 つのタイプのソース管理システムの変更を表示します。
- **変更されたファイル** 変更されたファイルまたは変更されたファイルを含むディレクトリ。ファイル (ファイル ビューの場合) をクリックすると、ファイルの詳細レポートが表示されます。ディレクトリ (ディレクトリ ビューの場合) をクリックすると、ソース管理システムの概要レポートが表示されます。

Files Updates

File	11/15/12	Total
xtest/common/com.parasoft.xtest.execution.api.web/classpath	1	1
xtest/common/com.parasoft.xtest.execution.api.web/src/com/parasoft/xtest/execution/api/web/runnable/WebFunctionalTestVisualizator.java	1	1
xtest/common/com.parasoft.xtest.results.api.web/classpath	1	1
xtest/common/com.parasoft.xtest.standards.api.web/classpath	1	1
xtest/common/com.parasoft.xtest.ws.web/src/com/parasoft/xtest/ws/web/workspace/WorkspaceUtil.java	1	1
xtest/eclipse/com.parasoft.xtest.changeadvisor.eclipse.ui.web/classpath	1	1
xtest/eclipse/com.parasoft.xtest.common.eclipse.ui.web/classpath	1	1
xtest/eclipse/com.parasoft.xtest.execution.eclipse.ui.soa/classpath	1	1
xtest/eclipse/com.parasoft.xtest.results.eclipse.core.soa/classpath	1	1

- **変更された行** 選択されたユーザーによって選択された日付に更新されたコード行の総数。括弧中の正の数字は、追加されたコードの行数を表します。括弧中の負の数字は、削除されたコードの行数を表します。

ファイル（ファイルビューの場合）をクリックすると、ファイルの詳細レポートが表示されます。ディレクトリ（ディレクトリビューの場合）をクリックすると、[ソース管理システムの概要]レポートが表示されます。

Lines changed

Directory	11/15/12	Total
xtest/common/com.parasoft.xtest.execution.api.web/	+1/-0	+1/-0
xtest/common/com.parasoft.xtest.execution.api.web/src/com/parasoft/xtest/execution/api/web/runnable/	+6/-3	+6/-3
xtest/common/com.parasoft.xtest.results.api.web/	+1/-0	+1/-0
xtest/common/com.parasoft.xtest.standards.api.web/	+1/-0	+1/-0
xtest/common/com.parasoft.xtest.ws.web/src/com/parasoft/xtest/ws/web/workspace/	+34/-22	+34/-22
xtest/eclipse/com.parasoft.xtest.changeadvisor.eclipse.ui.web/	+1/-0	+1/-0
xtest/eclipse/com.parasoft.xtest.common.eclipse.ui.web/	+1/-0	+1/-0
xtest/eclipse/com.parasoft.xtest.execution.eclipse.ui.soa/	+1/-0	+1/-0
xtest/eclipse/com.parasoft.xtest.results.eclipse.core.soa/	+1/-0	+1/-0
xtest/eclipse/com.parasoft.xtest.results.eclipse.core.virt/	+1/-0	+1/-0
xtest/eclipse/com.parasoft.xtest.standards.eclipse.ui.web/	+1/-0	+1/-0
xtest/test/com.parasoft.xtest.common.eclipse.ui.web.tests/	+1/-0	+1/-0
xtest/test/com.parasoft.xtest.junit.eclipse.core.web/	+1/-0	+1/-0
xtest/test/com.parasoft.xtest.standards.eclipse.ui.web.tests/	+1/-0	+1/-0
xtest/test/com.parasoft.xtest.testcases.eclipse.ui.web.tests/	+1/-0	+1/-0
xtest/test/com.parasoft.xtest.testcases.eclipse.ui.web.tests/src/com/parasoft/xtest/testcases/web/	+6/-3	+6/-3
Total	+59/-28	+59/-28

- 変更されたトークン** 選択されたユーザーによって選択された日付に変更されたトークンの総数。括弧中の正の数字は、追加されたトークンの行数を表します。括弧中の負の数字は、削除されたトークンの行数を表します。

ファイル（ファイルビューの場合）をクリックすると、ファイルの詳細レポートが表示されます。ディレクトリ（ディレクトリビューの場合）をクリックすると、[ソース管理システムの概要]レポートが表示されます。

Tokens Changed

Directory	11/15/12	Total
xtest/common/com.parasoft.xtest.execution.api.web/	+0/-0	+0/-0
xtest/common/com.parasoft.xtest.execution.api.web/src/com/parasoft/xtest/execution/api/web/runnable/	+2/-0	+2/-0
xtest/common/com.parasoft.xtest.results.api.web/	+0/-0	+0/-0
xtest/common/com.parasoft.xtest.standards.api.web/	+0/-0	+0/-0
xtest/common/com.parasoft.xtest.ws.web/src/com/parasoft/xtest/ws/web/workspace/	+124/-22	+124/-22
xtest/eclipse/com.parasoft.xtest.changeadvisor.eclipse.ui.web/	+0/-0	+0/-0
xtest/eclipse/com.parasoft.xtest.common.eclipse.ui.web/	+0/-0	+0/-0
xtest/eclipse/com.parasoft.xtest.execution.eclipse.ui.soa/	+0/-0	+0/-0
xtest/eclipse/com.parasoft.xtest.results.eclipse.core.soa/	+0/-0	+0/-0
xtest/eclipse/com.parasoft.xtest.results.eclipse.core.virt/	+0/-0	+0/-0
xtest/eclipse/com.parasoft.xtest.standards.eclipse.ui.web/	+0/-0	+0/-0
xtest/test/com.parasoft.xtest.common.eclipse.ui.web.tests/	+0/-0	+0/-0
xtest/test/com.parasoft.xtest.junit.eclipse.core.web/	+0/-0	+0/-0
xtest/test/com.parasoft.xtest.standards.eclipse.ui.web.tests/	+0/-0	+0/-0
xtest/test/com.parasoft.xtest.testcases.eclipse.ui.web.tests/	+0/-0	+0/-0
xtest/test/com.parasoft.xtest.testcases.eclipse.ui.web.tests/src/com/parasoft/xtest/testcases/web/	+3/-1	+3/-1
Total	+129/-23	+129/-23

プログラミング トークンについて

トークン (プログラミング) は、ソース コードの基本コンポーネントです。文字は、プログラミング言語の規則に従って、機能を表す 5 種類のトークン (定数、識別子、演算子、予約語、区切り子) に分類されます。

C 言語の次の式を使って考えてみましょう。

```
sum=3+2;
```

この行は、次の表で指定されているようにトークンに分けることができます

トークン	トークン タイプ
sum	indent
=	assign_op
3	number
+	add_op
2	number
;	semicolon

[ソース管理システムの概要] ページに表示されるのは、指定の日付に「編集された」プロジェクトファイルだけです。すべてのプロジェクト ファイルのリストについては 130 ページの「プロジェクトのソース コード」を参照してください。

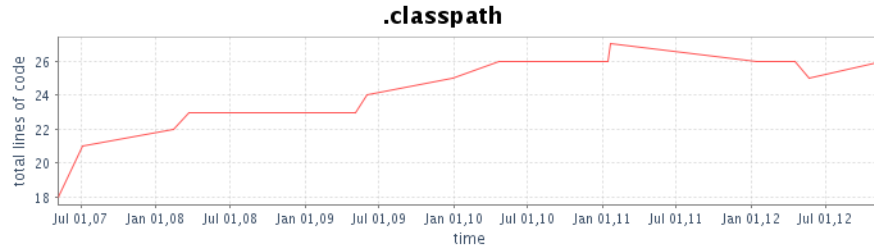
ファイルの詳細

[ファイルの詳細] ページには、ファイルの詳細テーブルには、選択されたファイルが作成された日からファイルに加えられたすべての改訂の履歴を含む、履歴情報が表示されます。選択されたファイルについて行われた改訂ごとに次の情報がリストされます。

- **ファイル** ファイルの名前。クリックするとソース コードが表示されます。
- **日付** 改訂が行われた日付と時刻。
- **バージョン** 改訂が属するバージョン。
- **ユーザー** ファイル中のコードを担当したユーザー。
- **変更された行** この改訂で変更されたコードの行数。
- **変更されたトークン** この改訂で変更されたトークンの数。
- **コメント** リストされた改訂に関連するコメント。

File Details

File: xtest/common/com.parasoft.xtest.execution.api.web/.classpath



Date	Version	User	Lines changed	Tokens Changed	Comment
11/15/12 4:43 AM	1.20	modtest	+1/-0	+0/-0	Task 20167: Module - Results
5/22/12 5:07 PM	1.19	jakubiak	+1/-2	+0/-0	@task 41135 - Enable dev mode on Monrovia projects.
4/17/12 5:25 PM	1.18	benken	+1/-1	+0/-0	@task 42996 - compile with Java 6
1/12/12 2:41 AM	1.17	dario	+0/-1	+0/-0	Removed unused reference to com.parasoft.xtest.sourcecontrol.project
1/19/11 5:35 PM	1.16	truong	+1/-0	+0/-0	@task 34009 - Add dependency on com.parasoft.xtest.filedefect
1/12/11 9:52 AM	1.15	jakubiak	+2/-2	+0/-0	Add back correct path to source folders
1/11/11 8:21 AM	1.14	maciek	+2/-2	+0/-0	Fix illegal link to source folder.
4/21/10 4:51 AM	1.13	marzec	+1/-0	+0/-0	task 12240: Fix Enh. 85697 : Put Source Control revision information in reports.
12/30/09 12:21 PM	1.12	zohrabb	+1/-0	+0/-0	PR 87340 @task 6555 - The "View Rule Documentation" option is broken for the Security "rules"
6/1/09 4:53 PM	1.11	jakubiak	+1/-0	+0/-0	PR 83703 - @task 3745 - Use WebInsureFunctionalTestViolation instead of InsureViolation in SOAtest
5/5/09 6:51 AM	1.10	maco	+1/-1	+0/-0	@task 11142 @PR 83318 : add libs.base project
4/22/09 10:44 AM	1.9	benken	+1/-1	+0/-0	@task 2519 - use J2SE-1.5 execution environment
4/21/09 5:03 PM	1.8	benken	+1/-1	+0/-0	@task 2519 - build Monrovia code with Java 1.5
3/13/09 7:05 PM	1.7	benken	+1/-1	+0/-0	PR 80047 - @task 2542 - use stax in xtest.libs
8/25/08 10:36 AM	1.6	benken	+3/-3	+0/-0	PR 76630 - rename webking folder to root
3/20/08 4:28 PM	1.5	jiu	+2/-1	+0/-0	add in direct dependency on stax-api.jar
2/12/08 11:53 AM	1.4	jehongm	+3/-2	+0/-0	PR 72156 - Dependence on results.api.web
7/3/07 4:45 PM	1.3	jakubiak	+3/-0	+0/-0	Added functional test violation type.
5/4/07 7:36 AM	1.2	marzec	+1/-1	+0/-0	Refactored.
5/4/07 7:25 AM	1.1	marzec	+18/-0	+0/-0	Added.

プロジェクトのソースコード

[プロジェクトのソースコード] ページを表示するには、コードベースのサイズレポートで [プロジェクトソースコードを表示] リンクをクリックします。[プロジェクトのソースコード] ページには、選択されたプロジェクトのメインディレクトリが表示されます。サブディレクトリとファイルを参照するには、ディレクトリをクリックします。[プロジェクトのソースコード] ページにはす

すべてのプロジェクト ファイルがリストされます ([ソース管理の概要] ページには指定の日付に「編集された」すべてのファイルが表示されます)。

Project Source Code	
Directory of:	
Directories:	39
Files:	0
Name ▲	Type
	[directory]
alex	[directory]
Attic	[directory]
bpelmaestro	[directory]
classes	[directory]
com	[directory]
config	[directory]
cpp	[directory]
dbc	[directory]
devadmin	[directory]
dtracer	[directory]
etc	[directory]

[プロジェクトのソース コード] ページには次の情報が表示されます。

- **Directory of** ユーザーがソース リポジトリ ツリー中のどこにいるかを表示します。上の例では、ルート ディレクトリの内容がリストされ、ディレクトリがまだ選択されていないため、このフィールドが空白になっています。
- **ディレクトリ** 選択されたディレクトリに含まれるサブディレクトリの数。
- **ファイル** 選択されたディレクトリにあるファイルの数。
- **名前** 選択されたディレクトリにあるサブディレクトリ / ファイルの名前。クリックすると、適切なファイルにたどりつくまで、ディレクトリの次のレベルが表示されます。ファイル名をクリックすると、[ファイルの詳細] ページが表示されます。詳細については 129 ページの「ファイルの詳細」を参照してください。
- **タイプ** リストされた名前がディレクトリなのかファイルなのかを示します。ファイルの場合はファイルの拡張子が表示されます。

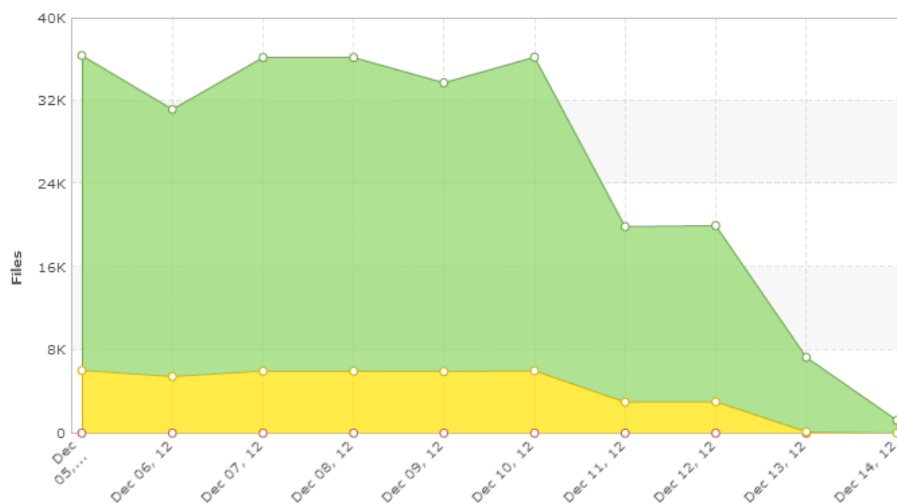
ビルド結果

レポート ビューで [プラクティス] メニューの [ビルド] をクリックすると、ビルド結果レポートが表示されます。ビルド結果ページの詳細については、62 ページの「ビルド レポート」を参照してください。

[ビルド結果] レポートには、リストされたドロップ日ごとに次のビルド情報が表示されます。

- 失敗したファイルの数
- 警告があり、未完了のファイルの数
- 成功したファイルとモジュールの数

Build Results



Date	Failed	Warning	Passed
Dec 14, 2012	0	21	1234
Dec 13, 2012	0	138	7174
Dec 12, 2012	0	3037	16938
Dec 11, 2012	3	2980	16909
Dec 10, 2012	1	6002	30198
Dec 09, 2012	2	5938	27784
Dec 08, 2012	2	5980	30176
Dec 07, 2012	2	5980	30184
Dec 06, 2012	4	5448	25709
Dec 05, 2012	3	6027	30320

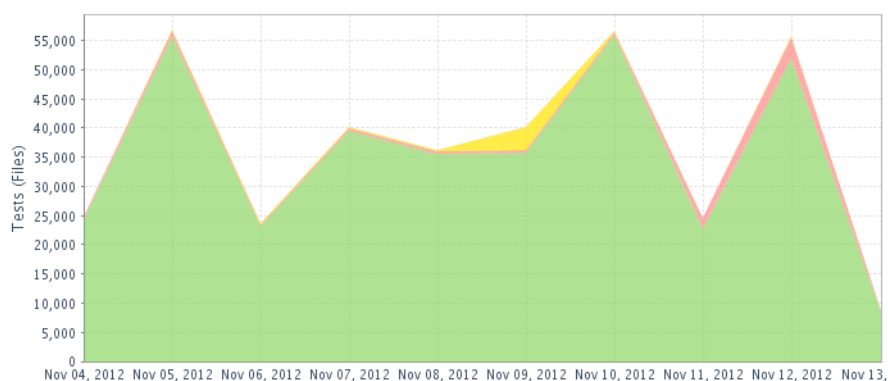
テスト (ファイル)

テスト (ファイル) レポートは、プロジェクトの現在の状態の概要を表示します。レポートには、指定された期間内に実行されたテストがリストされます。

1. レポート ビューで [テスト] メニューの [テストの概要] をクリックします。
2. [テスト (ファイル)] をクリックします。

各行は、特定の日に実行されたすべてのテストについて、未完了、失敗、成功のテスト ケースの数を示します。このテストには、手動テストを除き、静的解析、単体テスト、回帰テストなどが含まれます (手動テストの結果は [手動テスト セッション] のグラフに表示されます。手動テストの詳細については 149 ページの「手動テスト セッション」を参照してください)。

Tests (Files)



日付	Date	Number of Tests (Files)			Total	Duration	Execution Time Diff
		Incomplete	Fail	Pass			
+	Nov 13, 2012	0	252	8410	8662	3h 47m 43s	-3d 10h 11m 2s
+	Nov 12, 2012	237	3501	51968	55706	3d 13h 58m 45s	7h 45m 33s
+	Nov 11, 2012	0	1718	22763	24481	3d 6h 13m 11s	1h 27s
+	Nov 10, 2012	238	509	55827	56574	3d 5h 12m 43s	2d 13h 16m 48s
+	Nov 09, 2012	3936	507	35596	40039	15h 55m 55s	-1d 3h 14m 32s
+	Nov 08, 2012	237	408	35520	36165	1d 19h 10m 28s	3h 19m 17s
+	Nov 07, 2012	237	484	39424	40145	1d 15h 51m 11s	3h 46m 25s
+	Nov 06, 2012	229	122	23190	23541	1d 12h 4m 45s	-1d 8h 12m 57s

日付をクリックすると、静的解析および単体テストの結果が表示されます。

- 静的解析の場合、テストされたファイルの数が表示されます。
- 単体テストの場合、テスト (テスト スイート) ファイルが表示されます。

さらに、[テスト (ファイル)] レポートでは量と質の両方の面でテストを比較することができます。選択されたテスト グループに追加されたテストの数をチェックするだけでなく、テストのステータスが最近どのように変わったかも把握することができます。言い換えると、ステータスが " 失敗 " から " 成功 " (あるいはその逆) に変わったテストの数を参照できます。たとえば、前は成功だったテストが今回は失敗していることに気づいた場合、原因を調査して問題を解決できます。レポートからアクセスできる詳細情報が、問題の原因を発見するのに役立ちます。


詳細情報によって、指定の日付に実行された、プロジェクトの 2 つの別のテストを比較することができます。テストは、全般的な実行結果から 1 つのテスト メッセージの結果まで、さまざまなレベルで比較できます。Report Center は次のようにテストを比較します。

- 「ツール、マシン、解析タイプ」ごとのテスト実行。134 ページの「タイプ別テスト」を参照。
- 1 つの「ツール、マシン、解析タイプ」内のテスト実行。136 ページの「タイプ別テストの詳細」を参照。
- 1 つのテスト グループ内のテスト実行。138 ページの「テスト グループの詳細」を参照。

次のエンティティ ファクターが比較されます。

- 実行されたテスト実行、テスト グループ、またはテストの総数。
- 実行されたテスト実行、テスト グループ、またはテストの総数における、未完了 / 失敗 / 成功の数。
- テスト実行、テスト グループ、またはテストの累積的実行回数。

実行されたテストごとに、次の情報が表示されます。

- **ベースラインの管理** () 選択された日付の結果をベースラインとしてマークすることができます。ベースラインの結果は、テストされたソース コードが良い状態にあり、比較対象として機能することを表します (詳細については 141 ページの「ベースラインの設定」を参照)。
- **日付** テストが実行された日付。
- **未完了** テスト実行中に発生した予想外のエラーのために未完了になったテストの数。
- **失敗** テスト実行に失敗したテスト ケースの数。
- **成功** テスト実行に成功したテスト ケースの数。
- **合計** 選択された日付に実行されたテスト ケースの総数 (手動テストを除く)。
- **継続時間** 選択された日付に実行されたすべてのテスト ケースの累積実行時間。
- **実行時間 Diff** 前のテスト ケース実行と現在のテスト ケース実行との時間差。

テスト (ファイル) 詳細レポート

[テスト (ファイル)] レポートから、さらに別の統計を参照することができます。[日付]、[未完了]、[失敗]、[成功] の列のリンクをクリックすると、タイプ別テストレポートが表示されます

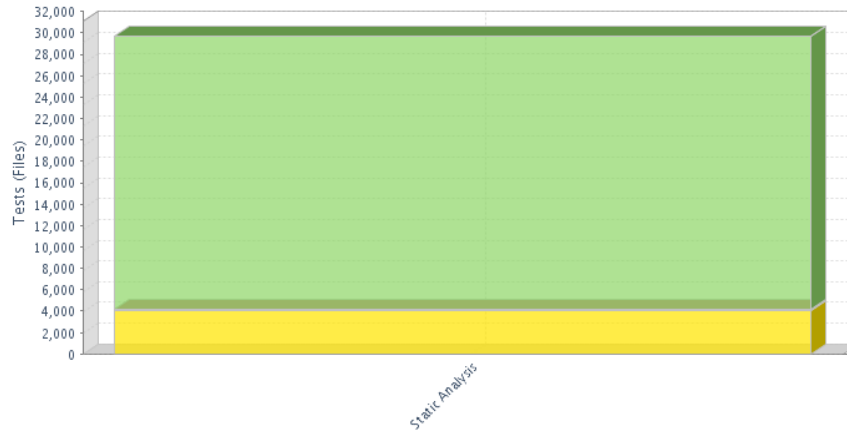
タイプ別テスト

[テスト (ファイル)] レポートを開き、日付または [未完了]、[失敗]、[成功] 列の値をクリックすると、[タイプ別テスト] レポートが開きます。

[タイプ別テスト] レポートは、テストが実行された「テスト タイプ、ツール、マシン」の異なる組み合わせごとに、テスト結果の比較を示します。各行には、[テスト] レポートの表でクリックされ

た日付について、「テスト タイプ、ツール、マシン」の異なる組み合わせごとに、未完了、失敗、成功の数が表示されます。この表は、テスト実行における量的変化と質的变化を示します。

Tests (Files) by Type



Test type	Machine	Tool	Number of Tests (Files)												Execution time		
			Incomplete			Fail			Pass			Total			Base	Curr.	Diff.
			Base	Curr.	New	Base	Curr.	New	Base	Curr.	New	Base	Curr.	Diff.			
Static Analysis	devel199	Jtest		229	229		47	47		15346	15346		15622				59m 40s
Static Analysis	anteater	Jtest	2	8	8	6	25	20	2716	6989	4264	2724	7022	4298	10m 11s	27m 12s	17m
Static Analysis	meadowlark	Jtest	3988	3995	8	9	9	5	3850	3917	74	7847	7921	74	7m 7s	7m 32s	25s
Total			3990	4232	245	15	81	72	6566	26252	19684	10571	30565	19994	17m 18s	1h 34m 25s	1h 17m 6s

[タイプ別テスト] ページには次の情報も表示されます。

- **ベース** 現在のテスト結果と比較したい日付のテスト結果。
- **現在** 現在の (最新の) テスト結果
- **新規** 「現在のテスト実行にはあるが、比較実行 (ベース) にはなかったテスト結果」か「ステータスが変わったテスト結果」のいずれか。
- **Diff** "ベース" と "現在" の結果の差。

量的な変化 (例)

量的な変化を説明するために、例を見てみましょう。[タイプ別テスト (ファイル)] レポートはテスト活動を示します。表中の数字は、おかしな点や改善が必要な点をユーザーが発見しやすいよう、異なるフォントで区別されます。

- **太字** 現在のデータを反映し、その日に解析が実行されたことを示します。太字のリンクのデータをクリックすると、さらに詳細なレポートが表示されます。
- **灰色の斜体** 現在の日付で実行されなかった解析を示します。これはデータを比較できないことを意味し、比較された日付の解析データだけが表示されます。
- **赤の斜体** 現在のデータとベースデータの比較において、解析が矛盾を発見したことを示します。言い換えると、徹底的に調べるべきである望ましくない状況があります。

[タイプ別テスト (ファイル)] レポートによって、ユーザーはソース コードに対する改善を見るだけでなく、変更がどのようにテストとアプリケーションの機能に影響しているかを見ることができます。

Test type	Machine	Tool	Incomplete			Fail			Pass			Total		Execution time			
			Base	Curr.	New	Base	Curr.	New	Base	Curr.	New	Base	Curr.	Diff.	Base	Curr.	Diff.
Functional Testing	thorn/10.9.1.193	JUnit	0	0	0	8	6	3	39	41	5	47	47	0	6h 2m 6s	4h 50m 44s	-1h 11m 22s
Static Analysis	gobos	Jtest	0	0	0	144	138	1	1182	1188	7	1326	1326	0	2h 55m 16s	2h 39m 50s	-15m 26s
Unit Testing	gobos	Jtest	0	0	0	25	23	1	736	737	4	761	760	-1	2h 36m 54s	2h 6m 54s	-29m 59s
Functional Testing	zuza/10.9.1.171	SOAtest	0	0	0	0	0	0	133	133	0	133	133	0	56s	1m 6s	9s
Total			0	0	0	177	167	5	2090	2099	16	2267	2266	-1	11h 35m 14s	9h 38m 35s	-1h 56m 38s

上の図の「テスト タイプ、ツール、マシン」を見てみましょう。JUnit はマシン Thorn で Functional Testing を実行しています。[合計] 列を見ると、現在のテスト実行 (現在 列) は設定されたベースライン (ベース 列) から変化がないことがわかります。ベースラインも現在もテスト実行の回数は 47 回です。比較された正確な日付は [タイプ別テスト] ページの右上に表示されます。

ここで、[失敗] および [成功] 列を見てみましょう。失敗したテストと成功したテストの数に違いがあることがわかります。これは、一部のテストのステータスが変化したことを意味します。その原因はアプリケーションのソース コードの変更またはテスト シナリオの変化です。

失敗したテストについて考えてみましょう。以前は 8 個のテストが失敗し (失敗 / ベース 列)、現在は 6 個のテストが失敗しています (失敗 / 現在 列)。ただし、そのうち 3 個は新しいテストです (失敗 / 新規 列)。新しい失敗したテストの以前のステータスは、"成功"、"未完了"、または "未実行" です。しかし、今回はステータスが "失敗" に変わりました。

計算してみましょう。

- $6-3=3$: 以前に失敗した 3 個のテスト ケースだけが現在も失敗しています。
- $8-3=5$: 以前に失敗した 5 個のテスト ケースが "成功" に変わったことを表します。おそらく、エラーのあったコードがいくつか修正され、その結果テスト ケースが成功したと思われる。
- 以前は 39 個のテスト ケースが成功していました。上記の計算から、新しく成功したテスト ケースは 5 個です。すなわち $39+5=44$ です。しかし、今回 "成功" のテスト ケースは 41 個だけです。
- $44-41=3$: 以前に成功した 3 個のテスト ケースが今回は失敗しています。これは新しい 3 個の失敗です。

[タイプ別テスト] レポートの表中のリンクをクリックすると、[タイプ別テストの詳細] レポートでさらに詳しいデータを参照できます。

タイプ別テストの詳細

[タイプ別テスト] レポートを開き、リンクをクリックすると、[タイプ別テストの詳細] レポートが開きます。

[タイプ別テストの詳細] の表に表示されるデータは、[タイプ別テスト] レポートの表で選択された「テスト タイプ、マシン、ツール」の詳細を反映します。基本的に [タイプ別テストの詳細] の表は、選択された「テスト タイプ、マシン、ツール」内でのテスト グループの実行を比較します。

Status: Incomplete Show All Tests Type | Machine | Tool: Static Analysis | devel199 | Jtest

Tests By Type Details

November 2012							December 2012						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1	2	3							1
4	5	6	7	8	9	10	2	3	4	5	6	7	8
11	12	13	14	15	16	17	9	10	11	12	13	14	15
18	19	20	21	22	23	24	16	17	18	19	20	21	22
25	26	27	28	29	30	1	23	24	25	26	27	28	29
							30						

Date	Group Name	Number of Tests (Files)									Execution time								
		Incomplete			Fail			Pass			Total								
		Base	Curr.	New	Base	Curr.	New	Base	Curr.	New	Base	Curr.	Diff.	Base	Curr.	Diff.			
2012-12-13 01:07	Coding Standards - WebKing			226	226			25	25			7540	7540			7791			59m 23s
2012-12-13 02:20	Coding Standards - WebKing			3	3			22	22			7806	7806			7831			16s
Total		0	229	229	0	47	47	0	15346	15346			15622						59m 40s

[タイプ別テストの詳細] レポートには次の情報も表示されます。


- **合計 (行)** 選択された「テスト タイプ、マシン、ツール」内にあるすべてのテストグループについてのサマリー データ。この行からさらに詳細な情報にアクセスできます。詳細については下記を参照してください。

注意! 必要に応じて、[タイプ別テストの詳細] の表のデータをこのページで直接フィルタリングすることができます。つまり、新しい「テスト タイプ、マシン、ツール」を選択するために [タイプ別テスト] レポートに戻る必要はありません。

- **グループ名** 表示されているテスト グループに割り当てられた名前。

[タイプ別テストの詳細] レポートで利用できるデータ フィルターは次のとおりです。

- **ステータス** 選択されたステータスのテスト結果だけを表示します。
- **新規ステータス** 新しいステータスがあるテスト結果だけを表示します (新しいステータスは、現在のデータとベース データとの間に矛盾が発見されたときに得られます)。
- **テスト タイプ / マシン / ツール** 選択された日付の「テスト タイプ、マシン、ツール」のテスト結果を表示します。
- **ベース / 現在** 選択された日付間のテスト結果を比較します。

各データ フィールドの右側の [カレンダー] アイコン  以外に、[タイプ別テストの詳細] ページの中央のカレンダーを使っても **ベース日** を選択することができます。カレンダーでは比較に使用できるデータがある日付の下には下線が表示されます。最大 2 ヶ月までのドロップが利用可能です。下線がある日付をクリックすると、現在の日付のデータと比較して結果を表示できます。灰色の日付はベースラインの日付を表します (141 ページの「ベースラインの設定」を参照)。

次のいずれかをクリックして [テスト グループの詳細] レポートを表示します。

- 特定のテスト グループの失敗したテスト ケースの数
- テスト グループ名 ([グループ名] 列)
- 失敗した新規テストの総数 ([合計] 行)

テスト グループの詳細

[タイプ別テストの詳細] レポートを開き、表のリンクをクリックすると、[テスト グループの詳細] レポートが開きます。

[テスト グループの詳細] レポートには、選択されたテスト グループについての詳細が表示されます。選択された結果と変更ステータスに基づいて、テスト グループのリストがフィルタリングされます。

異なる日の結果を比較することは、以前はなかったコードの不具合を発見するのに役立ちます。[ベース / 現在] のステータスを見ると、以前の結果と新しい結果の間に挿入されたコードが不正な動作を引き起こしていることが分かります。同様に、[継続時間] の急激な増加はパフォーマンスの問題 (減少している場合は最適化) を示している可能性があります。

[テスト グループの詳細] レポートに表示されるデータは、このレポートへのアクセス方法に基づいてソートされます。

この項目をクリック	表示される内容
[タイプ別テストの詳細] レポート： 特定のテスト グループの未完了 / 失敗 / 成功の テスト ケースの数	そのテスト グループについて実行されたテスト ケースで、選択されたステータスに一致するすべてのテスト ケース。
[タイプ別テストの詳細] レポート： 特定のテスト グループ名	そのテスト グループについて実行された、任意のステータスのテスト ケース。
[合計] に表示される未完了 / 失敗 / 成功のテストの総数	選択した「テスト タイプ、マシン、ツール」で実行されたすべてのテスト グループ。テスト グループ名ごとにプロジェクトの状況を包括的に眺めることができます。

SOAtest の結果 を除き、テスト グループの詳細は次のように表示されます。

Test Group Details						
gcc-2.95.3/C++TestRegress/xharness run: 2007-10-03 11:13						
Name	Base Runs	Base Status	Current Status	Base Duration	Current Duration	Duration Diff
src/case00002.cpp	0		Fail		406ms	
src/case00003.cpp	0		Fail		343ms	
src/case00004.cpp	0		Fail		343ms	
src/case00005.cpp	0		Fail		344ms	
src/case00006.cpp	0		Fail		360ms	
src/case00007.cpp	0		Fail		359ms	

[テスト グループの詳細] レポートには次の情報が表示されます。

項目	説明
名前	テストに割り当てられた名前。
ベースの実行	ベース テストの実行回数。
ベースのステータス	ベース テストのステータス - 未完了、失敗、成功のいずれか。
現在のステータス	現在のテストのステータス - 未完了、失敗、成功のいずれか。
ベースの継続	ベース テストの完了にかかった時間。
現在の継続	現在のテストの完了にかかった時間。
継続 Diff	ベース テストの完了にかかった時間と現在のテストの完了にかかった時間の差異。

ファイルは次のように色分けされます。

色	意味
緑	問題は修正されています。以前は (ベース日に) 発生していましたが、このレポートの対象の日付では発生しませんでした。
赤	新規の問題です。以前は (ベース日に) 発生しませんでした、このレポートの対象の日付では発生しました。

これらの結果に対して以下の操作を行うことができます。

- どのような問題がどこで発生したかを確認します。それには、テスト グループ名をクリックして [テストの詳細] レポートを表示します。プロジェクトに対して SourceScanner が実行されている場合、行番号のリンクをクリックしてソース コードを参照できます。
- ステータスで結果をフィルタリングします。
 - **任意**：すべてのテストが表示されます。
 - **任意のエラー**：現在のステータスが「失敗」または「未完了」のテストが表示されます。
 - **失敗、未完了、成功**：選択したステータスのテストが表示されます。
- 結果をフィルタリングして、すべてのテストを表示するか、または結果が変わったテストだけを表示します (たとえば「成功」から「失敗」に変わったり、「未完了」から「成功」に変わったテストです)。
- 2つのフィルタリング オプションを組み合わせると、特定の種類の結果を表示します。たとえば、「成功 / 変更を表示」を設定すると、ステータスが「失敗」から「成功」に変わったテストだけが表示されます。

SOAtest の結果

[タイプ別テストの詳細] レポートを開き、表の SOAtest でのテスト結果へのリンクをクリックします。

SOAtest の結果は、他の結果とやや異なった形で表示されます。

- テストスイートごとにテストがグループ化されます。図 51 のように、テストはツリー形式で表示されます。ツリーノードを展開したり縮小したりできます。
- テストスイートノードにすべての子テストの情報が集約されて格納されます。テストスイート情報がない場合、すべてのテストがルートノードの子になります。
- 1 個以上のテストが失敗した場合、テストスイートのステータスは「失敗」です。すべてのテストが成功した場合、ステータスは「成功」です。
- 継続時間は、すべての子テストの合計です。

SOAtest の結果に対して以下の操作を行うことができます。

- ツリーの矢印をクリックしてノードを展開 / 縮小します。
- 個々のテストの「ベースの状態」と「現在の情報」をクリックして、前回または現在のテストの詳細を参照します。

図 51: テストグループの詳細 : SOAtest の結果

	Base Runs	Base Status	Current Status	Base Duration	Current Duration	Duration Diff
task_2170_snake.tst (2011-01-05 11:38)	1	Fail	Fail	1s	1s	94ms
Test Suite	1	Fail	Fail	1s	1s	94ms
Functional Tests	1	Fail	Fail	1s	1s	94ms
task_2170	1	Fail	Fail	1s	1s	94ms
Test 5: Click "About" [Firefox 3.6]	1	Fail	Fail	1s	1s	94ms

(Firefox 3.6) Unable to perform user action: Element line>About not found

Copyright (c) 1996-2010 Parasoft

テストの詳細

[タイプ別テストの詳細] レポートを開き、名前をクリックすると、[テストの詳細] レポートが表示されます。

[テストの詳細] レポートには、選択されたテストについての情報が表示されるほか、ログに送られたすべてのメッセージの詳細リストが表示されます。

Test Details:

Summary Information

Date: 2006-02-25 13:56:53
 Test group name: Coding Standards
 Test ID: 63229370
 Test Name: AbstractEdgScanner.cc
 Test Coverage: -

All messages: 4

Messages:

Message ID	Message	Status	Error type	File	Line	User	%Coverage
76477605	Local variable p declared but not used. Avoid unus...	FAILED	Coding Violation	AbstractEdgScanner.cc	127	rozenau	-
76477606	Declare an assignment operator in class AbstractEd...	FAILED	Coding Violation	AbstractEdgScanner.h	53	darío	-
76477607	If a subclass implements a virtual function, use L...	FAILED	Coding Violation	AbstractEdgScanner.h	113	darío	-
76477608	If a subclass implements a virtual function, use L...	FAILED	Coding Violation	AbstractEdgScanner.h	122	darío	-



76477605 FAILED Local variable p declared but not used. Avoid unus...


Attribute	Value
status	failure
error_type	Coding Violation
line	127
test_category	Performance
severity	Severity Level 4
symbol	C:\home\twoflower\tomek\dw\cpp\codewizard\src\wizard\AbstractEdgScanner.cc
file	C:\home\twoflower\tomek\dw\cpp\codewizard\src\wizard\AbstractEdgScanner.cc
description	Avoid unused local variables or variables which have no side effect
rule_id	NosideEffectLocalVar.rule
message	Local variable p declared but not used.

ベースラインの設定

ベースライン オプションを利用すると、特定のレポート (選択された日のテスト結果) を「良い状態にあるテスト対象ソース コード」としてマークすることができます。ベースラインの目的は、今後のテスト結果と比較してプロジェクトの傾向を知るための基準を持つことです。

ベースラインの設定ダイアログを表示するには次のいずれかの操作を行います。

- [テスト] レポートで、表の 1 列目の先頭にあるベースライン アイコン  をクリックするか、その下のいずれかのプラス記号  をクリックします (適切な日付に対応するプラス記号を選択してください)。

[テスト] レポートからアクセスできる詳細レポート (タイプ別テスト , タイプ別テスト 詳細 , テスト グループ 詳細) で右上のベースライン アイコン  をクリックします。

Close

Add new baseline

Date

Description

Add

Existing baselines

Delete

<input type="checkbox"/>	Date	Description

ベースラインを設定するには、次の操作を行います。

1. ベースラインの設定ダイアログの [日付] フィールドで、今後のテスト結果と比較するためのベースとして設定する日付を指定します。
2. [説明] フィールドで、たとえば「For best results」のようにベースラインについての説明を入力し、[追加] ボタンをクリックします。

新しいベースラインが [既存のベースライン] の下に表示されます。

<input type="checkbox"/>	Date	Description
<input type="checkbox"/>	2007-10-01	best results for grs 2.7

3. 新しいベースラインのチェックボックスをオンにしてアクティブにし、[閉じる] ボタンをクリックします。
4. [タイプ別テストの詳細] レポートを開くかリフレッシュします。
ベースラインの日付が灰色で表示されます。

Tests By Type Details

September 2007							October 2007									
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat			
						1			2	3	4	5	6			
2	3	4	5	6	7	8	7	8	9	10	11	12	13			
9	10	11	12	13	14	15	14	15	16	17	18	19	20			
16	17	18	19	20	21	22	21	22	23	24	25	26	27			
23	24	25	26	27	28	29	28	29	30							
0 0 0 0							777 777 0 1151 1151									
Incomplete			Fail				Pass									
			Base			Curr.			New							
harness			0			0			367			367			0 0	
xharness			0			0			359			359			0 0	

5. ベースラインの日付をクリックします。
最新のテスト実行のデータとベースラインのデータが表示され、ユーザーは結果を比較することができます。

ベースラインを非アクティブにするには、次の操作を行います。

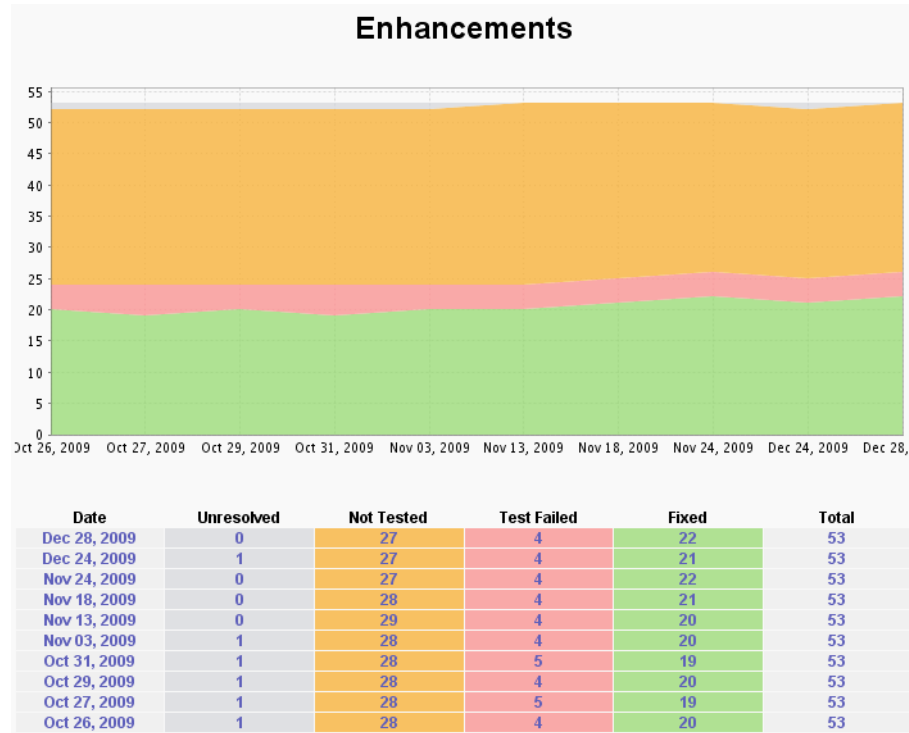
- ベースラインの設定ダイアログで、ベースラインの日付のチェックボックスをオフにし、[閉じる] ボタンをクリックします。

ベースラインを削除するには、次の操作を行います。

- ベースラインの設定ダイアログで、削除するベースラインの日付のチェックボックスをオンにして [削除] ボタンをクリックします。そして [閉じる] ボタンをクリックします。

欠陥および改善レポート

欠陥 / 改善 レポートには、特定の期間におけるプロジェクトの欠陥 / 改善のステータス（未解決、未テスト、失敗、修正済み）の傾向が表示されます。この情報は、プロジェクトの欠陥 / 改善に対する作業をモニタリングするのに役立ちます。



欠陥 / 改善 レポートから分かること

欠陥 / 改善 レポートは、プロジェクト ステータスについての重要な情報をレポートします。この情報はプロジェクトの品質のスナップショットであり、プロジェクトがどこでどのように進行しているかを示します。

たとえば、過去 12 週間におけるプロジェクトの欠陥 / 改善の状況が以下のようにになっている場合、その状況はそれぞれ次のことを意味します。

- 欠陥の総数は急増しているが、解決された項目はそのペースに追い付いていない：**
 欠陥の修正にもっと時間をかけるか、あるいはバグを防止するために何らかの措置を取った方が良いでしょう。
- 未テストの項目の数が徐々に増加している：**
 開発チームは、バグ トラッキング システム中の PR/FR を修正しているが、自動テストまたは手動テストを追加して作業の品質を確認していない。

詳細情報へのアクセス

各ステータスの列から、そのステータス（未解決 / 未テスト / 失敗 / 修正済み）に属する欠陥 / 改善のリストを参照することができます。

以下のページは、選択した日付とステータスについてレポートされたすべての欠陥 / 改善に加えて、名前、作成日、オーナーといった基本的なプロパティを表示します。

Daily Enhancements						
Any	Unresolved	Not Tested	Test Failed	Fixed		
ID	Created	Description	Assigned To	Status	Report Center Status	Test Required
90286	May 31, 2010	Make more obvious that task description is not editable	januszst@parasoft.com	NEW	Unresolved	✓
90136	May 19, 2010	Task Deletion	januszst@parasoft.com	NEW	Unresolved	✓
90095	May 17, 2010	Edit Scenario: assign to task	januszst@parasoft.com	NEW	Unresolved	✓
89795	Apr 23, 2010	Hierarchical projects	januszst@parasoft.com	NEW	Unresolved	✓
89553	Apr 07, 2010	Trivial: Update JIRA jars description	jhicken	VERIFIED	Not Tested	✓
89427	Mar 29, 2010	allow mapping of tests to tasks	januszst@parasoft.com	NEW	Unresolved	✓
89366	Mar 25, 2010	Support for RedHat 9.x or Higher, AS5	januszst@parasoft.com	RESOLVED	Not Tested	✓
89350	Mar 25, 2010	Trivial - note in email notification	jhicken	VERIFIED	Not Tested	✓
89307	Mar 23, 2010	Possibility to write comments to test execution	januszst@parasoft.com	NEW	Unresolved	✓
89213	Mar 17, 2010	Ability to delete scenario from Search scenarios result	januszst@parasoft.com	NEW	Unresolved	✓

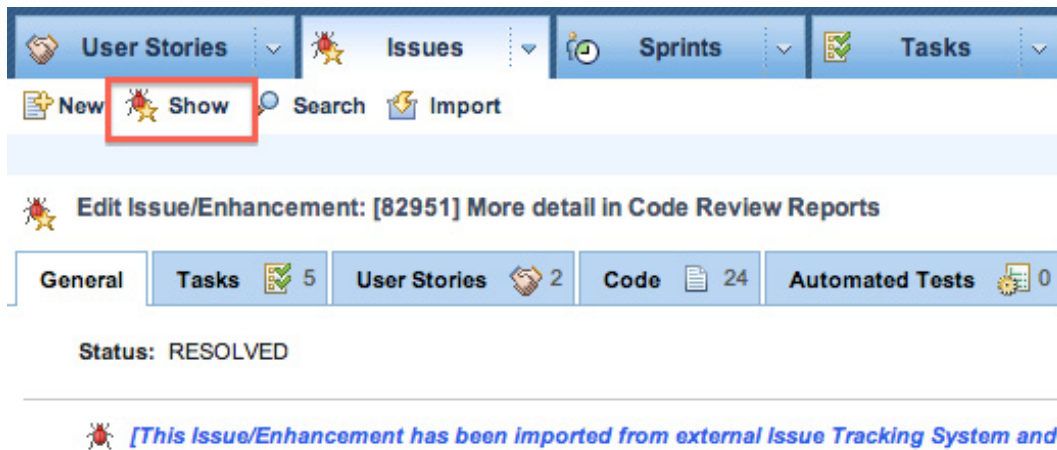
Defects Detailed Report						
Any	Unresolved	Not Tested	Test Failed	Fixed		
ID	Created	Description	Assigned To	Status	Report Center Status	Test Required
90313	Jun 02, 2010	LicenseServer GUI problem - 'token expired'	mali	RESOLVED	Not Tested	✓
90131	May 19, 2010	Concerto doesnot seem to be up , but is running	mali	NEW	Unresolved	✓
89908	Apr 30, 2010	Show source does not work on grs.parasoft.com.pl - svn	januszst@parasoft.com	NEW	Unresolved	✓
89863	Apr 28, 2010	Restest required feature does not work on SOAtest functional tests	januszst@parasoft.com	NEW	Unresolved	✓
89773	Apr 21, 2010	Japanese characters garbled in File Details page	sang	VERIFIED	Fixed	✓
89706	Apr 16, 2010	Export scenario to another project: aoutho overridden	januszst@parasoft.com	NEW	Unresolved	✓

詳細情報の操作

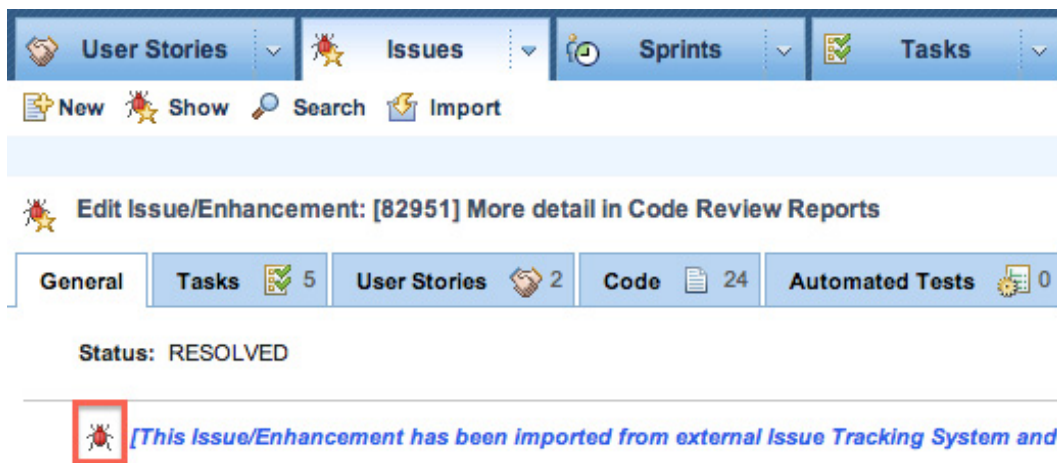
これらのページから次の操作を行うことができます。

- 欠陥 / 改善の ID 番号をクリックして、特定の欠陥 / 改善の詳細ページを開く

- 特定の欠陥 / 改善のバグ トラッキング システムのページを開く (JIRA や Bugzilla のようにバグ トラッキング システムに Web インターフェイスがある場合)



- [全般] タブのバグ アイコンをクリックし、バグ トラッキング システムから欠陥 / 改善のステータスを確認する



- Report Center のステータスを確認する。欠陥 / 改善のテスト ステータスが次のように示されます。

ステータス	色	説明
未解決	灰色	バグ トラッキング システムで欠陥 / 改善はまだ解決されていない。
未テスト	オレンジ	欠陥は解決されたが、まだテストされていない。
失敗	赤	欠陥は解決されてテストされた。テストは失敗した。
修正済み	緑	欠陥は解決されてテストされた。テストは成功した。

ステータスによるフィルタリング


表の上にあるリンクをクリックして、たとえば「解決されたがまだテストされていない欠陥 / 改善」など、特定のステータスの欠陥 / 改善だけを表示することができます。たとえば、失敗したテストをすべてチェックしてそれらのテストの詳細を知りたい場合、[失敗したテスト] のリンクをクリックします。







Daily Enhancements						
Any		Unresolved	Not Tested	Test Failed	Fixed	
ID	Created	Description	Assigned To	Status	Report Center Status	Test Required
87702	Dec 14, 2009	[McAfee] Automatic Audit/Comments about tasks changes	januszst@parasoft.com	RESOLVED	Test Failed	✓
87181	Nov 16, 2009	Support for TAGs	januszst@parasoft.com	RESOLVED	Test Failed	✓
85234	Aug 06, 2009	Remember sorting settings	januszst@parasoft.com	RESOLVED	Test Failed	✓
84917	Jul 17, 2009	More params in task search criteria needed	kenys	RESOLVED	Test Failed	✓
79810	Nov 14, 2008	Easy review of Defect status on iteration page.	pawelf	RESOLVED	Test Failed	✓
78915	Oct 17, 2008	Update GRS cache of bugzilla information after changing bugzilla PRs	pawelf	RESOLVED	Test Failed	✓
78910	Oct 17, 2008	Ability to modify multiple tasks at the same time	januszst@parasoft.com	RESOLVED	Test Failed	✓
78833	Oct 16, 2008	It should be possible to specify/review estimates/times spent in hours	kenys	RESOLVED	Test Failed	✓

さらに詳細なテスト実行の情報




特定の欠陥 / 改善の ID をクリックすると、テスト実行についてのさらに詳しい情報を参照することができます。




- 自動テストの実行についての情報は [自動テスト] タブに表示されます。




 **Edit Issue/Enhancement: [101383] [case#30188] Ability to switch back Code Review issue from 'Done' to 'To Fix'**






General | **Tasks**  1 | User Stories  0 | Code  20 | **Automated Tests**  15 | Scenarios  1 | Scenario Runs  0

Project: Xtest 9.5

Unit Testing **Jtest@phalanx**  0  0  5

Unit Testing - com.parasoft.xtest.codereview.tests **Passed** 2012-12-22 01:11  0  0  5

com.parasoft.xtest.codereview.internal.ReviewThreadStatusEvaluatorTest **Passed** 2012-12-22 01:11  0  0  5

- testReopeningWithOneReviewer() **Passed** [FR 101383, TASK 19346] 
- testReopeningWithTwoReviewers() **Passed** [FR 101383, TASK 19346] 
- testStatusWithOneReviewer1() **Passed** [FR 101383, TASK 19346] 
- testStatusWithOneReviewer2() **Passed** [FR 101383, TASK 19346] 
- testStatusWithOneReviewer3() **Passed** [FR 101383, TASK 19346] 

- 手動テストの実行についての情報は [シナリオ] タブに表示されます。

Edit Issue/Enhancement: [101143] Views and perspectives customizations for Virtualize

General Tasks 3 User Stories 1 Code 15 Automated Tests 0 Scenarios 2 Scenario Runs 0

2 Item(s) New Assign Unassign

ID	Name	Owner	Scenario Run Result	Run By	Testing History	Run
17824	UI - Verify customizations for Virtualize		2012-10-19 11:39	marzec		
17833	customization, Virtualize, hiding views		2012-06-01 15:49	marzec		

欠陥 / 改善のテストが必要ない場合

ある特定の欠陥 / 改善に対してテストが必要ない場合、その欠陥 / 改善の編集画面で [テストが必要] チェックボックスをオフにすることで「テストは不要」として欠陥 / 改善をマークすることができます。

オーナー: naga [表示](#) [削除](#)

監視者: admin [表示](#) [削除](#)

作成者: admin

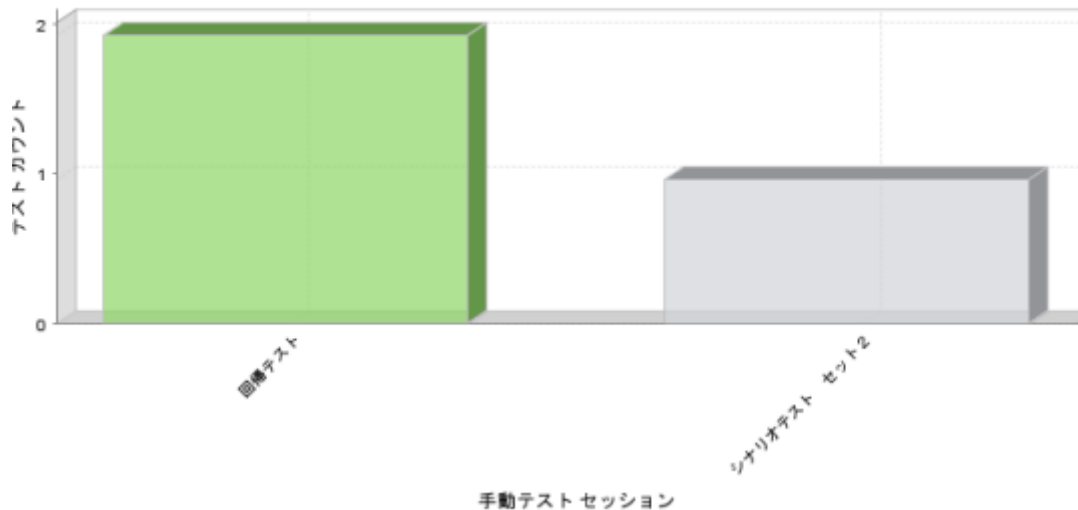
テストが必要:

ストーリー ポイント:

手動テスト セッション

レポート ビューで [テスト] メニューの [手動テスト セッション] をクリックすると、[手動テスト セッション] レポートが開きます。

[手動テスト セッション] レポートには、選択されたプロジェクトで実行または予定されたすべての手動テスト セッションが表示されます。このレポートのグラフは、テストの数とそのステータスを表します。



各テスト セッションの詳細は表に表示されます。各テスト セッションのテストに関してデータが表示されます。

Session Name	Start Date	Closure Date	Duration	Iteration ID	Planned Start Date	Deadline	#Not Run	#In Progress	#Fail	#Pass	#Incomplete	Rank
Concerto 3.5.1 - Basic Acceptance Tests before minor release	Dec 02, 2009	...	1d 2h 3m	417	Dec 02, 2009	Dec 03, 2009	0	0	0	4	0	
Concerto 3.5 release testing [2]	Nov 18, 2009	Nov 20, 2009	1h 33m				0	0	1	0	0	1
Concerto 3.5 release testing	Nov 02, 2009	Nov 20, 2009	12d 5h 34m		Nov 02, 2009	Nov 20, 2009	0	0	18	50	0	3
Scenario runs not belonging to any Manual Test Sessions							-	0	0	19	0	
Total:							0	0	19	73	0	

この表には次の情報が表示されます。

- セッション名** 各テスト セッションの名前。クリックすると [手動テスト セッションの編集] ページが開き、テスト セッションの詳細を編集できます。[手動テスト セッション] のスケジュールと編集の詳細については『Development Testing Platform Project Center ユーザーズ マニュアル』の「手動テスト セッション レポート」のセクションを参照してください。
- 開始日** テスト セッションが開始した実際の日付。
- 終了日** テスト セッションがクローズとしてマークされた日付。
- 継続時間** リストされたテスト セッションが進行中か完了したかによって、次のいずれかを意味します。
 - 進行中** これまでにセッションが実行された時間。

- **完了** セッションの完了までにかかった時間。
- **イテレーション ID** テスト セッションが属するイテレーションの ID。
- **開始予定日** テスト セッションを開始する予定日。
- **最終期限** テスト セッションを終了する予定日。
- **未実行** まだ実行されていない手動テストの数。
- **処理中** 進行中の手動テストの数。
- **失敗** 失敗した手動テストの数。
- **成功** 成功した手動テストの数。
- **未完了** 未完了の手動テストの数。
- **ランク** 手動テストの結果に基づいてテスト セッションに与えられた最終評価。

カバレッジ レポート

1. レポート ビューから [テスト] メニューの [テストの概要] をクリックします。
2. カバレッジ グラフをクリックしてカバレッジ レポートを開きます。

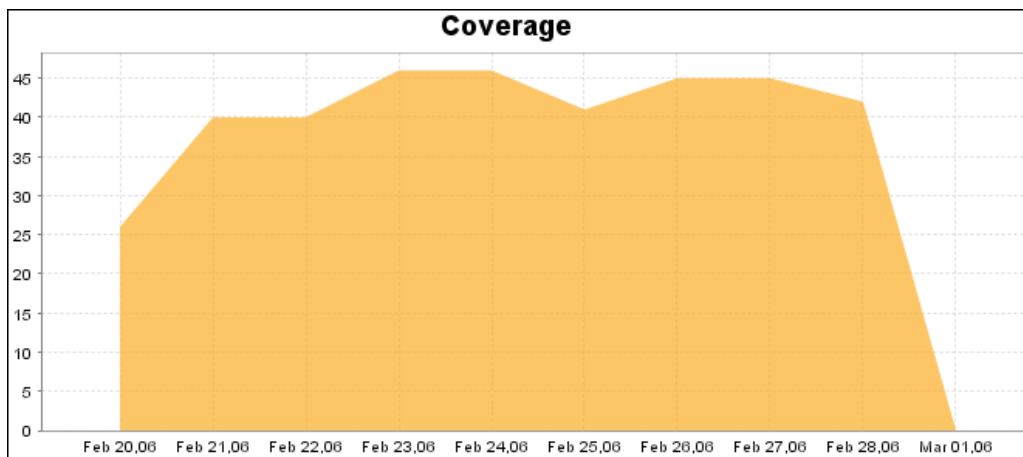
カバレッジの表には、次の情報が表示されます。

- コードの合計行数に対するテスト済みコード行数の割合
- 選択されたプロジェクトの単体テスト カバレッジ
- テスト済みのユニット、およびまだテストが必要なユニットの数

注意

- グラフには行カバレッジだけが表示されます。MC/DC やブランチ カバレッジなどは表示されません。
- カバレッジは、テスト対象として選択されたソース コード行数の合計に対する、テストが成功した行数のパーセンテージです。

カバレッジ レポートは、時間の経過に伴う単体テスト カバレッジの進捗をモニターするのに役立ちます。カバレッジのパーセンテージは徐々に増加するべきです。パーセンテージが増加しない場合、それはテストが作成されていないことを表します。



カバレッジ グラフの下にはカバレッジの表があります。この表からアーキテクトはカバレッジ率が改善しているかどうかを知ることができます。

Date	Unit Tests Coverage
Feb 25, 2008	35% (32800 / 94055)
Feb 24, 2008	38% (46833 / 123234)
Feb 23, 2008	38% (46629 / 123419)
Feb 22, 2008	39% (48939 / 124743)
Feb 21, 2008	38% (44739 / 118361)
Feb 20, 2008	51% (14674 / 28886)
Feb 18, 2008	39% (19366 / 50291)
Feb 17, 2008	42% (32919 / 78771)
Feb 16, 2008	43% (35645 / 82239)
Feb 15, 2008	41% (31061 / 74998)

カバレッジの表には、リストされたドロップ日ごとに次の情報が表示されます。

- カバレッジ率
- テストされたソース コードの行数
- テストの総数 (テスト済みと未テストの両方)

カバレッジの詳細レポート

カバレッジの表で日付をクリックすると、カバレッジの詳細レポートが開きます (Jtest、dotTEST、および Emma の結果のみ)。

Coverage Details

Package Name ^	Unit Tests Coverage (Tested Lines/Lines to Test)
examples.stackmachine	87% (369 / 425)

パッケージ内のクラスのカバレッジを参照するには、パッケージ名をクリックします。

Coverage Details

Package Name: examples.stackmachine

Class Name ^	Unit Tests Coverage (Tested Lines/Lines to Test)
AbstractStackMachine	100% (25 / 25)
CommandLineStackMachine	98% (58 / 59)
CustomListRenderer	100% (6 / 6)
FifoStackMachine	100% (7 / 7)
LifoStackMachine	100% (7 / 7)
RunnableStackMachine	89% (231 / 261)
RunnableStackMachine\$AddButton	75% (6 / 8)
RunnableStackMachine\$DivideButton	75% (6 / 8)
RunnableStackMachine\$FifoRadioButton	56% (5 / 9)
RunnableStackMachine\$ImageButton	100% (9 / 9)
RunnableStackMachine\$LifoRadioButton	60% (6 / 10)
RunnableStackMachine\$MultiplyButton	75% (6 / 8)
RunnableStackMachine\$PopButton	75% (6 / 8)
RunnableStackMachine\$PushButton	75% (6 / 8)
RunnableStackMachine\$PushTextField	57% (4 / 7)
RunnableStackMachine\$SubtractButton	75% (6 / 8)
StackList	58% (35 / 60)

メソッドごとの単体テスト カバレッジを参照するには、クラスをクリックします。

Coverage Details

Class Name: RunnableStackMachine

Method Name ^	Unit Tests Coverage (Tested Lines/Lines to Test)
componentHidden(ComponentEvent)	0% (0 / 1)
componentMoved(ComponentEvent)	100% (1 / 1)
componentResized(ComponentEvent)	100% (2 / 2)
componentShown(ComponentEvent)	100% (1 / 1)
createApiPanel()	100% (6 / 6)
createInputFieldsPanel()	100% (10 / 10)
createInputPanel()	100% (21 / 21)
createOperationButtonsPanel()	100% (24 / 24)
createPushPanel()	100% (17 / 17)
createRadioOptionsPanel()	100% (21 / 21)
createStackButtonPanel()	100% (16 / 16)
createStackPanel()	100% (17 / 17)
main(String[])	100% (4 / 4)
RunnableStackMachine()	100% (29 / 29)
windowActivated(WindowEvent)	100% (1 / 1)
windowClosed(WindowEvent)	0% (0 / 1)
windowClosing(WindowEvent)	0% (0 / 2)

Policy Center (レガシー) ユーザーズ ガイド

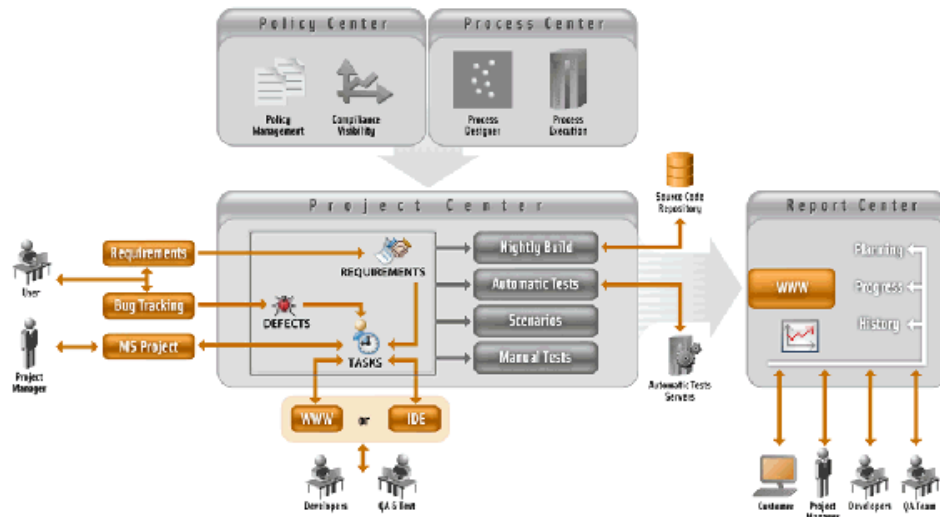
この章では、Parasoft Development Testing Platform に付属の Policy Center テクノロジーについて説明します。なお、Standard Edition にはない追加機能を持った Advanced Edition の Policy Center をダウンロードしてインストールすることもできます。詳細については Parasoft 製品の担当者にお問い合わせください。

- Policy Center の概要
- Policy Center (Standard Edition) との接続
- ポリシー チェックとレポートの設定
- ポリシー チェック結果の参照
- ポリシーの設定

Policy Center の概要

Policy Center テクノロジーは、ソフトウェア開発プロジェクトの全体的な状況を自動的にモニタリングするためのシステムです。このテクノロジーは条件を定義し、個々の条件はソフトウェア開発プロセスの側面を計測します。各条件は 1 つ以上のソースからのデータを取り込み、計測の結果、修正措置が正当化されるかどうか、そしてどの程度まで正当化されるかを判断します。最後に、Policy Center は全般的なスコアを決定します。そして、非常に単純かつ直接的なフォーマットですべての結果を電子メールで担当者に送信します。

Policy Center のプロセスは、ソース管理システム、バグ トラッキング システム、レポートング システムなど、まったく異なるトラッキング システムからソフトウェア ライフ サイクルについての情報を収集します。そしてプロジェクト固有の設定とメトリクスを適用して、毎朝プロジェクトの全体的な状況を判断します。



Policy Center (Standard Edition) との接続

Policy Center のライセンスをお持ちの場合、Advanced Edition をダウンロードしてインストールするか、Standard Edition をアクティブ化できます。Policy Center のライセンス取得方法および Advanced Edition についての詳細は、営業担当者にお問い合わせください。

Policy Center (Standard Edition) をアクティブ化するには、次の操作を行います。

1. `<DTP_HOME>/conf/ExternalServicesConfig.xml` 構成ファイルを開きます。
2. `<policyCenter>` 要素に次の行を追加します。

```
<url>/grs/jsf/policy/project_policy.jsf</url>
```

ポリシー チェックとレポートの設定

設定が完了すると、Policy Center は指定されたポリシーを指定された間隔でチェックし、指定された担当者に電子メールでレポートを送信します。さらに、ポリシー チェックの結果を Report Center のマネージャー ダッシュボードに送ります。

チェックとレポートを自動化するには、以下の設定が必要です。

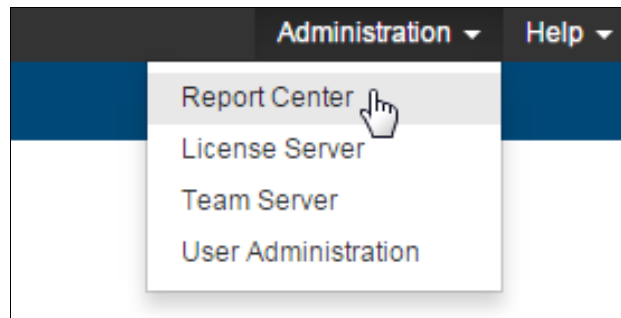
- 電子メール レポートの受信者の設定
- プロジェクト ポリシーのチェックとレポートの設定

自動化したチェックに加えて（あるいは自動化したチェックの代わりに）、好きなタイミングで手動でポリシー チェックを実行することもできます。詳細については 159 ページの「手動でのポリシー チェックの実行」を参照してください。

電子メール レポートの受信者の設定

Policy Center を使用する前に、ポリシー コンプライアンス状況の電子メール レポートを受け取るユーザーを指定する必要があります。次の操作を行います。

1. [管理] > [Report Center] をクリックします。



[設定] > [E-mail] を選択し、電子メールの設定を行います。

E-mail Settings

The screenshot shows the 'E-mail Settings' configuration interface. It is divided into several sections:

- SMTP Server Settings:** Includes fields for 'Server' (pigeon.parasoft.com), 'Port' (25), 'Connection Security' (None), 'User Name', 'Password', 'Sender' (concerto-daemon@snake.parasoft.), and 'Realm'. A note indicates that the Realm field is required for SASL authentication.
- Test the settings:** Features an 'E-mail address' input field and a 'Check' button. A note states that the test message will be sent to the specified e-mail.
- E-mail address domain:** Includes a 'User Default Domain' field set to parasoft.com. A note explains that this is used when no email is defined for a user.
- E-mail notification:** Contains a checked checkbox for 'Send e-mails when Project Center tasks are changed'.
- A 'Save' button is located at the bottom of the form.

2. 電子メールの設定をまだ行っていない場合、SMTP サーバーの設定を入力します。
3. [保存] ボタンをクリックします。

プロジェクト ポリシーのチェックとレポートの設定

Policy Center のチェックとレポートを有効化するには、次の操作を行います。

1. Policy Center で、[ポリシー] メニューの [プロジェクト ポリシー] をクリックします。
[プロジェクトのポリシー] ページが表示されます。

図 52: プロジェクトのポリシー

プロジェクトのポリシー

プロジェクト ポリシーの有効化:

ALM ポリシー
プロジェクト設定^{OFF}
 予算分析
 スケジュール分析
 機能分析
 アプリケーションのセキュリティ
 機能検証
 欠陥トレンド改善
 コード解析
 ビルド結果
 回帰テスト
 テストカバレッジ
 欠陥トレンド
 ソースコードトレンド
 ピアレビュー
 実行済み単体テスト

プロジェクト 設定 ?

 ヘルスチェック ターゲット日 (今日からのオフセット):
 グループ E-mail リスト:

 解析後アクション
 成功:
 許容範囲:
 失敗:

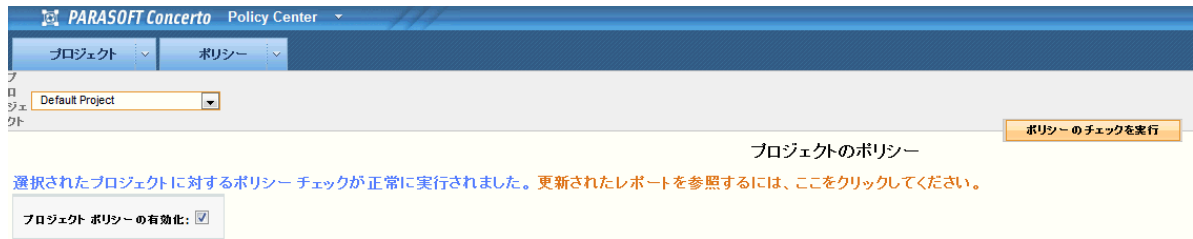
 ヘルスチェック実行日
 日曜 月曜 火曜 水曜 木曜 金曜 土曜

2. [プロジェクト ポリシーの有効化] チェックボックスをオンにします。
3. 170 ページの「ポリシーの設定」で説明しているように、ポリシーとその他の設定を行います。電子メールを生成して送信する曜日を必ず指定してください。
4. [保存] ボタンをクリックします。

手動でのポリシー チェックの実行

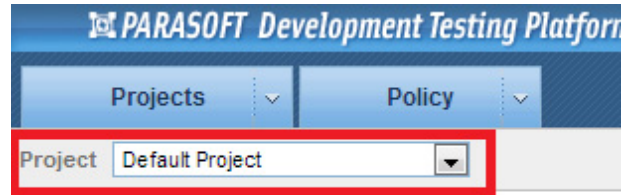
ポリシー チェックは、ユーザーの設定に基づいて実行されます。しかし、[ポリシーのチェックを実行] ボタンをクリックすると、選択したプロジェクトに対していつでも好きなタイミングでポリシー チェックを実行できます。ポリシー チェックを実行するには、次の操作を行います。

1. [ポリシーのチェックを実行] ボタンをクリックします。
2. ポリシー チェックが正常に完了すると、ステータス メッセージと共に最新のマネージャー レポートへのリンクが表示されます。リンクをクリックしてマネージャー レポートを参照できます。

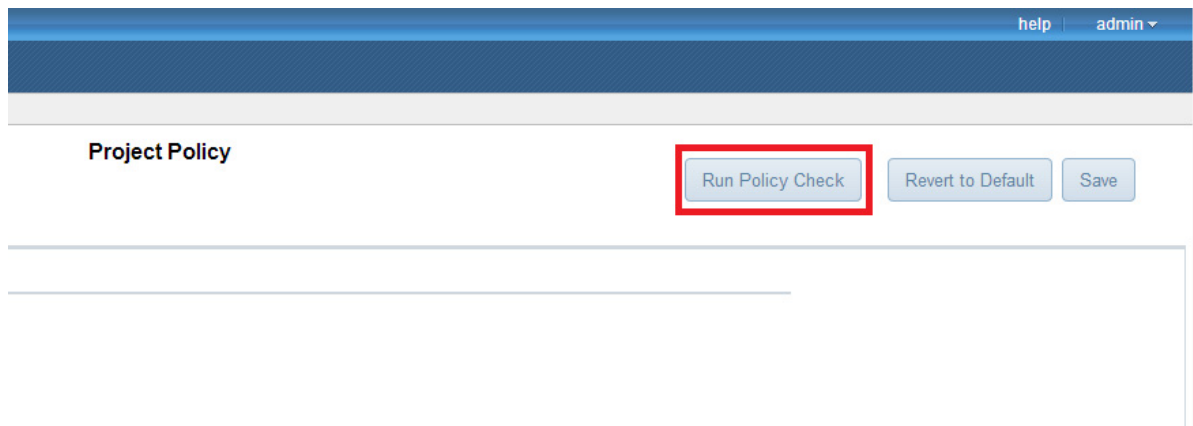


ポリシー チェック結果の参照

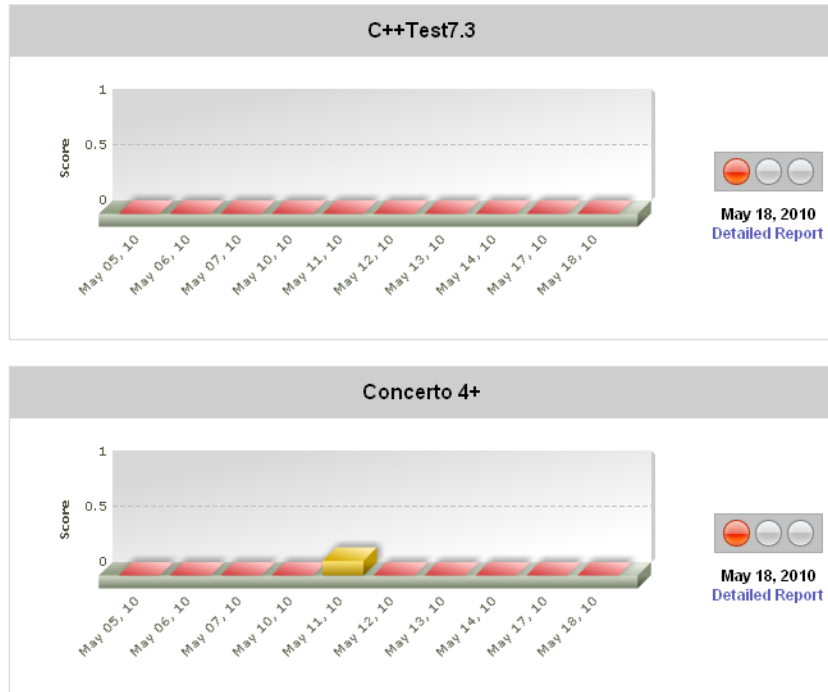
1. [Project] ドロップダウン メニューからプロジェクトを選択します。



2. [ポリシーのチェックを実行] ボタンをクリックします。実行には時間がかかる場合があります。



3. [更新されたレポートを参照するには、ここをクリックしてください。] リンクをクリックして、ポリシー チェックの結果を参照します。



電子メール レポート

適切な設定によって、マネージャーと担当チーム メンバーは、すべてのプロジェクトのステータスを電子メールで受け取ることができます。予算、スケジュール、および機能という点から各プロジェクトのステータスがレポートされます。

PARASOFT We make software work.		POLICY CENTER	
Manager Policy Check for Tuesday, January 4, 2011			
Project	Report Date	Status	
BPELMaestro	1/3/11		
Concerto 10	1/3/11		
Concerto 4+	1/3/11		
Concerto UI Review	1/3/11		
DotTest HEAD	1/3/11		
Insure 7.1	1/3/11		
Jtest 9.0	1/3/11		
SOAtest 9.1	1/3/11		
dotTEST 9.0	1/3/11		

結果への対応

これらのレポートに対する推奨する作業プロセスは以下のとおりです。

- ステップ 1: 全体的なプロジェクトヘルス レポートを参照する
- ステップ 2: ヘルシーではないプロジェクトについて、プロジェクト全体の最終期限、予算、機能のリスクをレビューする
- ステップ 3: レポートされたリスクごとに、イテレーション レベルでの詳細を確認する
- ステップ 4: 品質リスクがあるプロジェクトについて、プロジェクト レベルの詳しいヘルス チェックをレビューする
- ステップ 5: 適切な対応を決定する

ステップ 1: 全体的なプロジェクトヘルス レポートを参照する

すべての項目が黄信号または青信号である場合、プロジェクトはスケジュール、予算、機能の面で順調です。マネージャーは何もアクションを取る必要がありません。

赤い信号がある場合、マネージャーは [詳細レポート] リンクをクリックして、予算分析、最終期限分析、および機能完了分析を参照する必要があります。

このデータからマネージャーは「例外によるマネージメント」を実行できます。つまり、期待またはポリシーに反する項目だけを確認できます。

ステップ 2: ヘルシーではないプロジェクトについて、プロジェクト全体の最終期限、予算、機能のリスクをレビューする

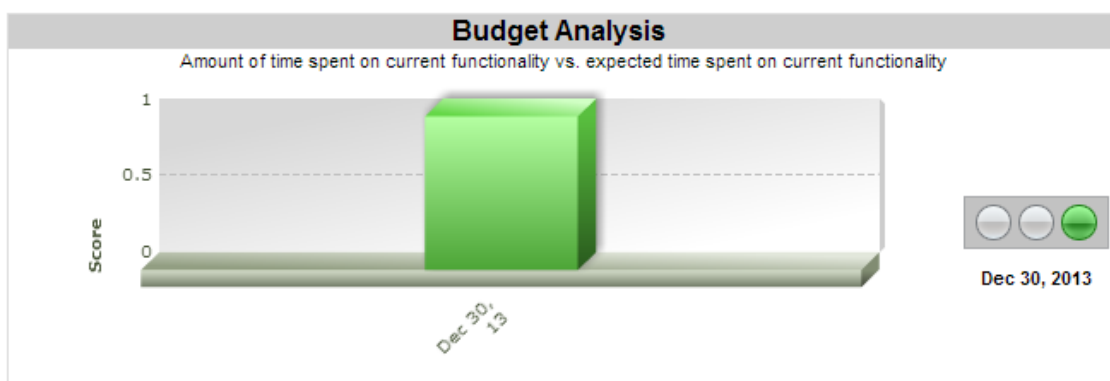
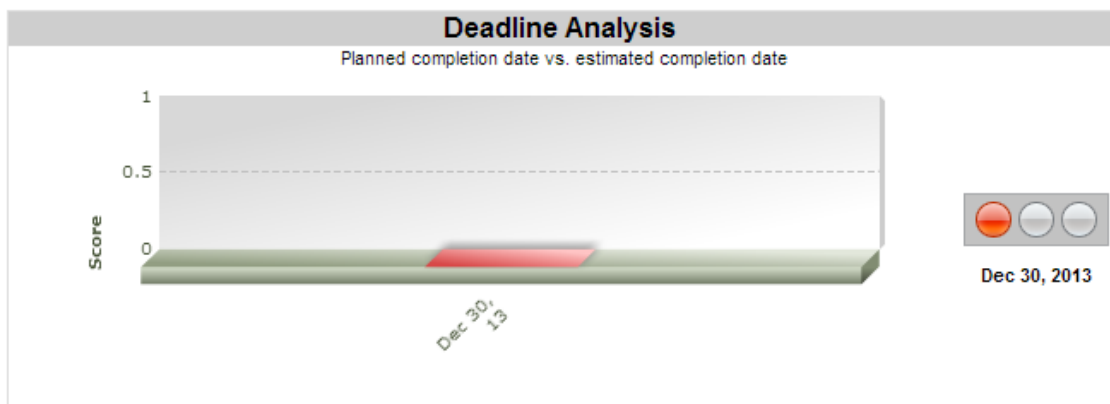
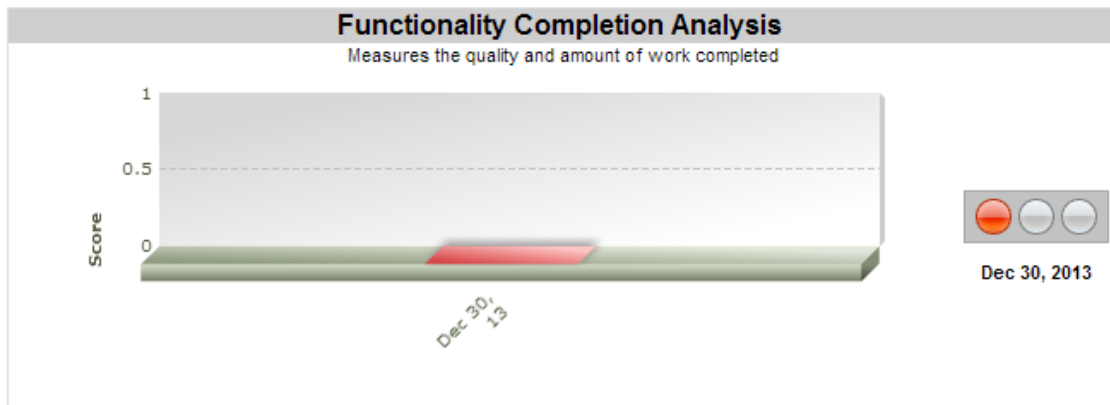
最上位のレポートで信号の箇所をクリックすると、そのプロジェクト中のどの部分にリスクが存在するかを示すダッシュボードが表示されます。

Project Dashboard

Default Project



Dec 30, 2013



一目で、マネージャーは効果的なプロジェクト管理のために必要な客観的な回答をリアルタイムで得ることができます。

- プロジェクトの機能は適切に実装されているか、その機能は動作するか（機能完了分析）

- プロジェクトは期限までに間に合うか (最終期限分析)
- プロジェクトは予算内に収まっているか (予算分析)

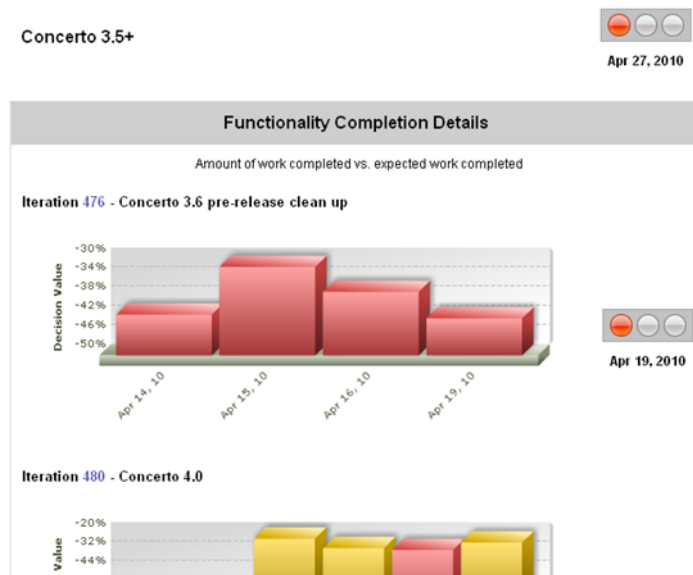
その後マネージャーは、ステップ 3 とステップ 4 で説明するように、イテレーションごとにさらに詳しい情報を参照し、リスクのある変動要素を確認できます。ステップ 5 で説明するように、どのように問題に対応すればよいかについて、詳細な情報を得た上で判断できます。

なお、予算と最終期限の信号よりも多くの情報を機能の信号が持つことに注意してください、これは重要です。機能完了分析は、予定される機能を実装するためのタスクが完了したかどうかを考慮するだけではありません。品質の欠陥から守るために、マネジメントによって選択された複数のポリシーに従って機能が実装されているのかも確認します。たとえば、マネージャーは Web インターフェイスを使って、少なくとも一定の割合のプロジェクトのソース コードをチームのテストケースが実行しなければならないことを指定できます。

ステップ 3: レポートされたリスクごとに、イテレーション レベルでの詳細を確認する

プロジェクト レベルでの予算、最終期限、機能の赤信号に対応するダッシュボード部分をクリックすると、詳細ページが表示されます。どのイテレーションにリスクがあるか、なぜリスクがあるのかについての詳しい情報がイテレーションごとにブレークダウンされて示されます。この時点で表示されるページは、クリックされたスコア (予算、最終期限、または機能) によって異なります。さまざまなページについては 167 ページの「ポリシー チェック レポートの確認」で説明します。

Functionality Details



ステップ 4: 品質リスクがあるプロジェクトについて、プロジェクト レベルの詳しいヘルスチェックをレビューする

前のレベルの Functionality Details レポートで作業品質の問題がレポートされた場合、赤、黄、緑でカラー表示されたレポート中のアイテムをクリックします。[ポリシー チェックの詳細] レポート

が開き、さらに詳しい情報にマネージャーは簡単にアクセスすることができます。たとえば、許容外の作業品質スコアがレポートされたのかを判断したり、プロジェクト レベルのすべてのヘルス インジケータに照らして情報をレビューします。

Default Project - Oct 21, 2012: 🔴🔵🟢

ステータス
リンク

予算分析

ID	名前	現在のコスト	現在の見積りコスト	決定値
1	Milestone 1: Architecture design and core framework implementation	0 日	0 日	0.0%
2	Milestone 2: Basic functionality implementation	0 日	5 日	-99.96%
3	Milestone 3: Advanced functionality implementation and release	0 日	0 日	0.0%

- Lower Bound ~ Upper Bound: 10% ~ 30%
- Positive Rate == Over-Budget, Negative Rate == Under-Budget

スケジュール分析

ID	名前	終了予定日	見積り終了日	決定値
1	Milestone 1: Architecture design and core framework implementation	2012-09-15	2012-10-31	150.0%
2	Milestone 2: Basic functionality implementation	2012-10-16	2012-11-15	100.0%
3	Milestone 3: Advanced functionality implementation and release	2012-11-16	2012-11-27	30.0%

- Lower Bound ~ Upper Bound: 0% ~ 30%

機能分析

ID	名前	完了済み作業 %	完了予定作業 %	決定値
1	Milestone 1: Architecture design and core framework implementation	37%	216%	-92.86%

プロジェクトのリスク解析

🔴🔵🟢

セキュリティ - アプリケーションのセキュリティ

- There is no activity on 2012-10-21.

実装 - 機能検証

ID	名前	名前
1	Milestone 1: Architecture design and core framework implementation	<ul style="list-style-type: none"> テスト変なし: 0% 失敗したテストの割合: 0% テストしきい値なし: 10% 失敗したテストのしきい値: 10%
2	Milestone 2: Basic functionality implementation	<ul style="list-style-type: none"> テスト変なし: 0% 失敗したテストの割合: 0% テストしきい値なし: 10% 失敗したテストのしきい値: 10%
3	Milestone 3: Advanced functionality implementation and release	<ul style="list-style-type: none"> テスト変なし: 0% 失敗したテストの割合: 0% テストしきい値なし: 10% 失敗したテストのしきい値: 10%

実装 - 欠陥レンド改善

- Checked Date: 2012-08-24
- # of Unresolved defects: 4 defects (80%)
- Threshold: 10%

実装 - コード解析

- There is no activity on 2012-10-21.

実装 - ビルド結果

- There is no build information from 2012-10-21.

ステップ 5: 適切な対応を決定する

この時点で、レポートされた問題にどのように対処するかについて、マネージャーは提供された情報に基づいた上で決断することができます。ポリシーまたは可変のしきい値が不適切だと感じた場合、マネージャーは GUI 上でポリシーまたはしきい値を変更できます。たとえば、予算の問題がレポートされたときに、予算分析をもっと緩くしたい場合、マネージャーは関連する条件を調整できます。

また、静的解析違反のポリシーを厳しくし過ぎたために機能の問題がレポートされているのであれば、そのポリシー設定を調整できます。

そうでなければ、つまりプロジェクトが本当に危機に瀕していると思う場合、マネージャーはどうやってプロジェクトを立て直すかを決定する必要があります。ソフトウェア開発のマネージメントでは妥協は避けられません。たとえば、予算分析が赤信号であるものとします。この場合、マネージャーには次の 2 つの選択肢があります。

- 機能を減らす (つまり、完了しなければならない作業の量を減らす)
- 予算を増やす (プロジェクト メンバーを増員することで、残りが作業をより早く完了する)

最善の解決策はマネージャーが何を優先するかによって変わります。重要ではない機能を発見できる場合、1 番目の選択肢が最善の方策かもしれません。また、予定した期限までにすべての機能を完了させなければならない場合、予算を増加するのが多分もっとも確実な方法でしょう。

しかし、固定費で契約していて予算に制限がある場合であれば、マネージャーは予算内で収まるように機能と期限を管理する方法を考える必要があります。また、予算の割り当てが徐々に増える場合 (プロジェクトの継続中に数回にわたって支払いがある場合)、マネージャーはそれに応じてプロジェクトを計画する必要があります。

ポリシー チェック レポートの確認

電子メール レポートとマネージャー ダッシュボード レポートは、同じ方法でポリシー準拠の結果をレポートします。

全体的なステータス

信号のアイコン (赤、黄、青) によって各プロジェクトの全体的なステータスが示されます。このステータスは、プロジェクトの予算、期限、および機能の分析に基づいて決定されます。



Oct 16, 2012
Detailed Report

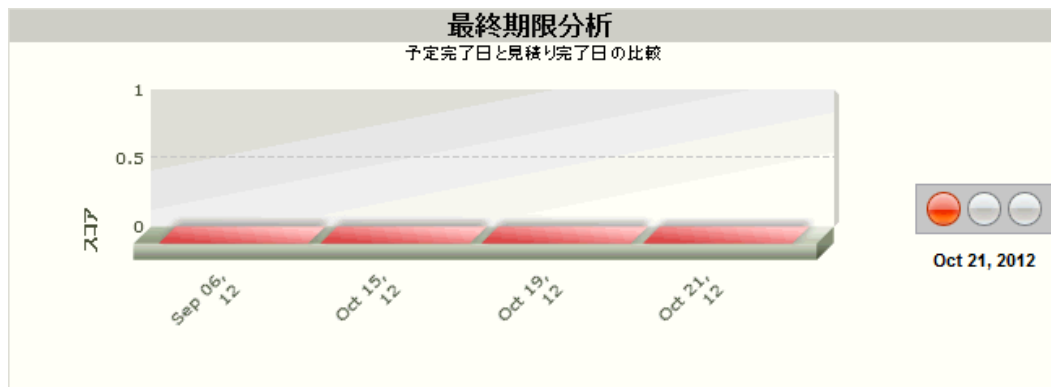
これらの信号のアイコンは、Policy Center レポートのすべての信号アイコンと同じように、0 から 1 の変数スコアに対応します。青信号は、常にスコア 1 を表します。赤信号はスコア 0 を表します。黄色はスコアが 0 と 1 の間であることを表します。測定されるすべての変数について、許容できるスコアまたは許容できないスコアのデフォルト値があらかじめ設定されています。170 ページの「ポリシーの設定」で説明するように、これらのデフォルト値はユーザーが変更できます。

最終期限分析

最終期限分析は、現時点でプロジェクトの進捗が遅れていないか、またはレポート日に遅れている危険性がないかをレポートします。

プロジェクトが計画どおりに進捗しているか、超過限度割合によって定義された適正なしきい値内に収まっている場合、青信号が表示されます。170 ページの「ポリシーの設定」で説明するように、このしきい値はユーザーが指定できます。

最初に見積られたプロジェクト期限の正確さを高めるために、プロジェクトの進捗状況についてのリアルタイムのフィードバックが利用されます。



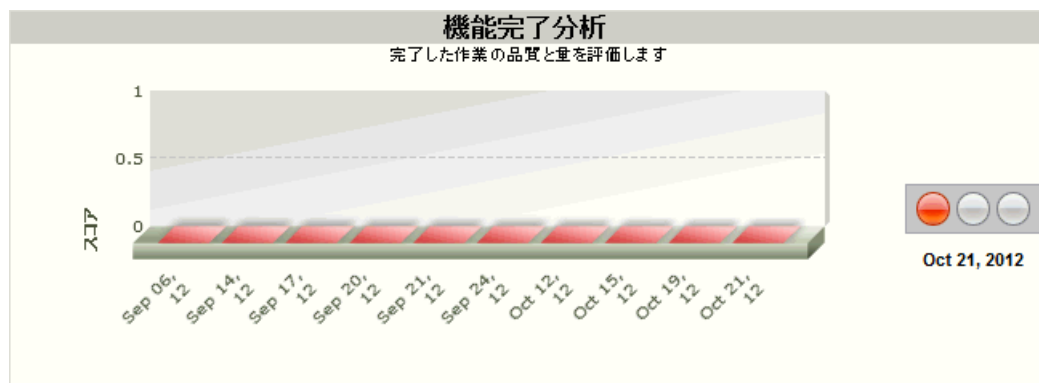
機能完了分析

機能完了分析レポートは、予定された機能の実装が完了しているかどうか、指定された品質ポリシーに準拠しているかどうかを示します。

このレポートは、「実際に完了した作業量」と「期待される完了予定の作業量」を対比します。

計画された機能の実装が遅れているか、またはレポート日の時点で遅れている危険性がある場合、警告が通知されます。計画された機能がスケジュールどおりに完成しているか、超過限度割合によって定義された適正なしきい値内に収まっている場合、青信号が表示されます。170 ページの「ポリシーの設定」で説明するように、このしきい値はユーザーが指定できます。

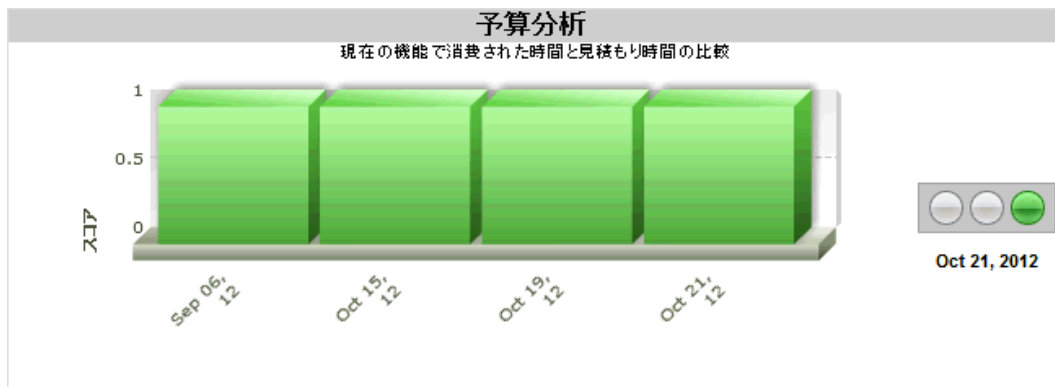
「静的解析の実行」や「単体テストの実行」といった指定されたポリシーにチームが準拠していない場合、機能が実際に適切に実装されていないのではないかという疑いが持ち上がります。



予算分析

このレポートは、現時点でプロジェクトが予算を超過していないか、またはレポート日に予算を超過している危険性がないかをレポートします。

青信号は、プロジェクトが予算どおりであるか、適正なしきい値（パラメータライズ可能）内に収まっていることを表します。170 ページの「ポリシーの設定」で説明するように、このしきい値はユーザーが指定できます。



ポリシーの設定

ポリシーは、プロジェクトがどの程度良く進んでいるかを表す基準です。Development Testing Platform では各ポリシーのしきい値をユーザーが設定できます。

Policy Center の [ポリシー] メニューから次の設定を行うことができます。

- マネージャー E-mail 設定：ポリシー チェックが実行された後にマネージャー レポートのメールを受け取るユーザー リストを定義できます。
- グローバル ポリシー：ポリシーごとにデフォルト値を設定できます。
- プロジェクト ポリシー：Policy Center レポートを有効化したり、各プロジェクトに対してポリシーを設定することができます。プロジェクトのポリシー ステータスについて Policy Center からマネージャーに電子メールを送信することもできます。

Policy Center の設定

マネージャー レポートに表示されるデータは、[プロジェクトのポリシー] ページで各ポリシーに設定された値に基づいて計算されます。各ポリシーにはあらかじめデフォルト値が設定されていますが、環境に合わせて最適な値に変更することを推奨します。また、各ポリシーを有効化 / 無効化することもできます。

Policy Center では以下のプロジェクト ポリシーを有効化して設定できます。

- プロジェクト設定
- 予算分析
- スケジュール分析
- 機能分析
- アプリケーションのセキュリティ
- 機能検証
- 欠陥の傾向 / 改善
- コード解析
- ビルド結果
- 回帰テスト
- テスト カバレッジ
- 欠陥の傾向
- ソース コード トレンド
- ピア レビュー
- 実行済み単体テスト

GUI 上のインジケーター

- アスタリスク (*) は、変更された値を表します。
- 斜体は、そのポリシーが無効化されていることを表します。

プロジェクト設定

[プロジェクト設定] は、プロジェクト ポリシーの全般的な設定を指定します。いつポリシーをチェックするのか、およびポリシー チェックの完了後にどんなアクションを実行するのかを指定します。

設定	説明
ヘルスチェック ターゲット日	<p>プロジェクトは、バックグラウンド ジョブによって毎日チェックされます。この "Health Check Back End" ジョブはデフォルトで午前 7 時に実行されます。</p> <p>[今日からのオフセット] は、どの日のデータをポリシーチェックの対象とするかを指定します。</p> <p>たとえば -1 に設定すると、Policy Center は前日のデータを探します。たとえば、火曜日の午前 7 時に実行された場合、月曜日のプロジェクト データ (テスト結果、最終期限 / 予算) が分析されます。0 に設定すると、今日のデータを探します。</p>
グループ E-mail リスト	<p>ポリシー チェックの実行が終了すると、このリストで指定されたユーザー管理グループのメンバー全員に電子メールが送信されます。</p> <p>ユーザー管理グループの詳細については 714 ページの「ユーザーの設定と管理」を参照してください。</p> <p>複数のグループを指定するには、「MyProjectA, MyProjectB, MyProjectC」のようにカンマで区切って指定します。</p>

設定	説明
解析後アクション	<p>ポリシー チェックが完了した後にサーバーが実行する 1 つ以上のコマンド (たとえば .bat/.sh スクリプト) を指定します。</p> <p>解析後のアクションを指定するには、[解析後アクション] チェックボックスをオンにし、適切なフィールドでコマンドを指定します。</p> <ul style="list-style-type: none"> • 成功 : 特定のポリシー チェックの結果が " 成功 (青信号)" だった場合にコマンド (スクリプト) を実行します。 • 許容範囲 : 特定のポリシー チェックの結果が " 許容可能 (黄信号)" だった場合、コマンド (スクリプト) を実行します。結果が " 許容範囲 " になるのは、定義された下限と上限の間に結果が入る場合です。 • 失敗 : 特定のポリシー チェックの結果が " 失敗 (赤信号)" だった場合にコマンド (スクリプト) を実行します。結果が " 失敗 " になるのは、1 つ以上のポリシー ファクターが " 失敗 " ステータスである場合です。 <p>コマンド (スクリプト) はサーバーによって実行されるため、コマンド (スクリプト) がこのマシンで実行できることが重要です。</p>
ヘルスチェック実行日	ポリシー チェックがレポートを生成する曜日。

予算分析

予算分析は、現時点でプロジェクトが予算を超過していないか、またはレポート日に予算を超過している危険性がないかをレポートします。プロジェクトが予算どおりであるか、適正なしきい値 (パラメータライズ可能) 内に収まっている場合、青信号が表示されます。

設定	説明
下限	予算しきい値の下限 (パーセント)。このしきい値を下回っていれば、順調としてレポートされます。
上限	予算しきい値の上限 (パーセント)。このしきい値を超えると、問題としてレポートされます。

Development Testing Platform では、資金ではなく日数でプロジェクトの予算とコストが計算されます。

初回の予算分析は、反復中の個々のタスクに対する作業見積りの合計に基づきます。Development Testing Platform にタスクを追加するときに、チーム メンバーがそのタスクの作業量を見積りします。Development Testing Platform はこの情報を組み合わせてプロジェクト全体の見積りを決定します。作業時間の合計の見積りは、予定されたプロジェクト期間全体に線形関数として分配されません。

チームメンバーがタスクの作業を実施する間、Development Testing Platform は各タスクに実際にかかった時間を計測します。Development Testing Platform はこのデータを見積りと比較して、反復が予算内に収まっているかどうかを判断します。プロジェクトの現時点に対して見積られた作業時間に収まっていない場合、この問題をマネージャーに警告するために赤信号が使用されます。

Policy Center の予算計算にどのタスク タイプを含めるかを選択することもできます。タスク タイプを選択するには、タスク タイプに対応するチェックボックスをオンにします。タスク タイプの説明については 440 ページの「タスクの新規作成」を参照してください。

スケジュール分析

スケジュール分析は、現時点でプロジェクトの進捗が遅れていないか、またはレポート日に遅れている危険性がないかをレポートします。プロジェクトが計画どおり進捗しているか、超過限度割合によって定義された適正なしきい値内に収まっている場合、青信号が表示されます。

設定	説明
下限	期限超過の下限（パーセント）。このしきい値を下回ってれば、順調としてレポートされます。
上限	期限超過の上限（パーセント）。このしきい値を超えると、問題としてレポートされます。

下限と上限の間の値は、黄信号と見なされます。スケジュール分析は、分析日と比較して " 予測される最終日 " が未来である反復だけを分析します。

機能分析

機能分析は、計画された機能のステータスをレポートします。計画された機能の実装が遅れているか、またはレポート日の時点で遅れている危険性がある場合、警告が通知されます。計画された機能がスケジュールどおりに完成しているか、超過限度割合によって定義された適正なしきい値内に収まっている場合、青信号が表示されます。

設定	説明
下限	タスク完了の下限（パーセント）。このしきい値を下回ると、問題（期限超過）としてレポートされます。
上限	タスク完了の上限（パーセント）。このしきい値を超えていれば、順調（予定通り）としてレポートされます。
猶予期間しきい値	現在のタスクの完了を問題としてレポートする前に必要なタスク完了期待値（パーセント値）です。猶予期間しきい値を超えた場合にだけ警告がレポートされます。

プロジェクト分析は、プロジェクトの反復ごとに、「実際に完了した作業量」と「完了が期待される作業量」を比較します。反復が赤信号である場合、プロジェクト レベルでも赤信号になります。

「完了した作業」は次のように計算されます。

$$\text{完了した作業の費用見積りの合計} / \text{すべての期待される完了した作業} * 100$$

「完了が期待される作業」は次のように計算されます。

$$\text{すべての期待される完了した作業} / \text{反復の日数} * \text{反復が開始してから経過した日数}$$

アプリケーションのセキュリティ

アプリケーションのセキュリティ レポートは、KLOC あたりの検出されたセキュリティ違反が特定の範囲を超えた場合にレポートされます。レポート日の時点でセキュリティ違反がパラメーターで指定された KLOC あたりの違反数を超えた場合、警告が通知されます。KLOC あたりのセキュリティ違反が許容範囲内にある場合、青信号が表示されます。

設定	説明
下限	1 KLOC あたりのセキュリティ違反数の下限。
上限	1 KLOC あたりのセキュリティ違反数の上限。

機能検証

機能検証は、レポート日の時点でテスト ケースがない、またはテストが失敗している「開発タスク」の割合をモニターします。テスト スイートが不足または失敗しているタスクの割合に基づいて警告が通知されます。不足または失敗しているテスト スイートの割合が許容範囲内である場合、青信号が表示されます。

設定	説明
テストがないタスクの割合 (%)	テストケースのないタスクの割合の上限 (%)
テストが失敗したタスクの割合 (%)	テストが失敗したタスクの割合の上限 (%)

欠陥の傾向 / 改善

欠陥の傾向 / 改善レポートは、テストが失敗している、またはテストがない「欠陥」の割合をモニターします。テストスイートが不足または失敗している欠陥の割合に基づいて、あるいはテストスイートが不足または失敗している欠陥の数が限度を超えた場合に警告が通知されます。不足または失敗しているテスト スイートの割合が許容範囲内である場合、青信号が表示されます。

設定	説明
テストが失敗した欠陥の割合 (%)	テストケースが失敗した欠陥の割合 (%)
受け入れられる最小欠陥数	許される欠陥数合計の最小値
テストがない欠陥の割合 (%)	テストケースがない欠陥の割合 (%)

この分析の目的は、個々の欠陥に少なくとも 1 個のテスト ケースが関連付けられていること、そして夜間テスト中にテストが成功することを保障することです。欠陥に関連付けられたテストが失敗する場合、それは重大なレグレッションが発生し、早急にレビューすべきであることを意味します。

欠陥に関連付けられたテストがない場合、コードの変更によって再び欠陥が発生したかどうかを自動的にチェックすることはできません。これは、優先順位 2 位 (黄色) の課題としてレビューする必要があります。

未解決の欠陥の割合がしきい値よりも大きい場合、レビューする必要があります (黄色)。

コード解析

コード解析レポートは、KLOC あたりの検出されたコード解析違反が特定の範囲を超えた場合にレポートされます。レポート日の時点でコード解析違反がパラメーターで指定された KLOC あたりの違反数を超えた場合、警告が通知されます。また、レポート日に発見された新しい違反の数が上限を超えた場合も警告が通知されます。KLOC あたりの違反が許容範囲内にある場合、青信号が表示されます。

設定	説明
1 日に受け入れられる新規違反数	1 日あたりの新規違反数の上限
下限	1KLOC あたりの違反数の下限
上限	1KLOC あたりの違反数の上限

ビルド結果

ビルド結果レポートは、ビルド プロセスからの警告メッセージをモニターし、プロジェクトで警告メッセージのあるファイル数の割合がしきい値を超えた場合に通知します。警告メッセージのないファイルの割合が適正な範囲内に収まっている場合、青信号が表示されます。

設定	説明
受け入れられる警告メッセージの割合 (%)	プロジェクトで許容される警告メッセージのレポートされたファイルの割合

回帰テスト

回帰テストレポートは、「テストの失敗」の割合が上限および下限の範囲内に収まっているかをモニターします。指定されたテストの失敗の割合を超えている場合、警告が通知されます。失敗の割合が許容範囲内である場合、青信号が表示されます。

設定	説明
下限	テスト失敗の割合の下限 (%)
上限	テスト失敗の割合の上限 (%)

テスト カバレッジ

テスト カバレッジ レポートは、プロジェクトの単体テストの行カバレッジが上限および下限の範囲内に収まっているかをモニターします。テスト カバレッジが低すぎる場合、警告が通知されます。カバレッジの割合が許容範囲内である場合、青信号が表示されます。

設定	説明
下限	テスト カバレッジの割合の下限 (%)
上限	テスト カバレッジの割合の上限 (%)

欠陥の傾向

欠陥の傾向レポートは、プロジェクトに混入した欠陥が上限および下限の範囲に収まっているかをモニターします。欠陥の合計に対して、新たに検出された欠陥の数が多すぎる場合、警告が通知されず。

設定	説明
下限	新規欠陥数 / 新規欠陥数の下限
上限	新規欠陥数 / 新規欠陥数の上限

この分析は、「前日に混入した新しい欠陥の総数」を「同じ日に修正された欠陥の総数」で割った値をチェックします。割合が上限よりも大きい場合、それは修正された欠陥よりも発生した欠陥の方が多いことを意味します。プロジェクトの欠陥が徐々に増えている可能性があり、プロジェクトをレビューすべきであることを表します。

ソースコードトレンド

ソースコードトレンドレポートは、開発チームの進捗を表す概観的な指標です。コードの合計行数に対して、削除されたコード行数が多すぎる場合、警告が通知されます。また、指定された日付範囲内にソースコードリポジトリにコミットされたコードがない場合にも警告が通知されます。適正な間隔でソースコードリポジトリにコードがコミットされており、コードの合計行数が増えている場合、青信号が表示されます。

設定	説明
変更行の下限	対象日と前日のコード行数の差分の下限 (%)
変更行の上限	対象日と前日のコード行数の差分の上限 (%)
アクティビティ日付の下限	最終コミット日の許容範囲下限 (日数)
アクティビティ日付の上限	最終コミット日の許容範囲上限 (日数)

ピアレビュー

ピアレビューレポートは、期待通りにピアコードレビューが実行されているか、またピアレビューで発見された問題が修正されているかをモニターします。特定のメンバーがタスクリストに抱える待機レビュータスクが多すぎる場合、問題のあるタスクの数が許容範囲を超えた場合、またはクローズされていないタスクの合計数が許容範囲を超えた場合に警告が通知されます。ピアレビューが期待通りに実行されている場合、青信号が表示されます。

設定	説明
最大タスク数の下限	許容されるタスクの最大数の下限
最大タスク数の上限	許容されるタスクの最大数の上限
課題数の下限	問題があるタスク数の下限 (レビューアーがコード中に問題を発見した)
課題数の上限	問題があるタスク数の上限 (レビューアーがコード中に問題を発見した)
未解決件数の下限	クローズされていないタスク数の下限 (レビューアーはまだ課題をレビューしていない)
未解決件数の上限	クローズされていないタスク数の上限 (レビューアーはまだ課題をレビューしていない)

未解決のタスクの数が「未解決件数の下限」よりも少ない場合、あるいは課題のあるタスクの数が「課題数の下限」よりも少ない場合、青信号が表示されます。

未解決のタスクの数が「未解決件数の上限」よりも多い場合、あるいは課題のあるタスクの数が「課題数の上限」よりも多い場合、赤信号が表示されます。

課題の割合は次のように計算されます。

「未解決の割合」が、「未解決の割合のしきい値の下限」よりも小さく、かつ「課題の割合」が「課題の下限」よりも小さい場合、「本日のレビューの総数」が「許容される未解決の課題」よりも少ないかどうかをチェックします。

未解決のコード レビュー タスクの割合は次のように計算されます。

$$\left(\text{本日のレビュー総数} (\text{"レビュー依頼"} \text{ と } \text{"開発者への修正依頼"} \text{ を含む}) - \text{昨日のレビュー総数} \right) / \text{昨日のレビュー総数} * 100$$

実行済み単体テスト

実行済み単体テスト レポートは、プロジェクトに関連付けられたすべての単体テストの実行状況および単体テストの失敗の割合をモニターします。実行済み単体テスト数の合計の割合が、過去 10 ドロップのテストにわたって許容範囲を下回った場合、またはテストケースの失敗の割合が許容範囲を超えた場合に警告が通知されます。単体テストが期待通りに実行されており、失敗が許容範囲内に収まっている場合、青信号が表示されます。

設定	説明
テストケース差異数の下限	過去 10 回のテスト中の合計テストケース数差異の下限 (%)
テストケース差異数の上限	過去 10 回のテスト中の合計テストケース数差異の上限 (%)
失敗したテストの割合の下限	失敗したテストケースの割合の下限 (%)
失敗したテストの割合の上限	失敗したテストケースの割合の上限 (%)

管理者

- DTP のインストール
- Development Testing Platform の設定
- Development Testing Platform の統合
- 付録 : Development Testing Platform 管理者ガイド

プロジェクトの作成と構成

Project Center と Report Center でのほとんどのアクティビティは、特定のプロジェクトに関連して実行されます。DTP ユーザー マニュアルの「プロジェクトの作成および設定」章では、プロジェクトに最も関係の深いアクションが説明されています。マニュアルを参照するには、DTP の UI で [ヘルプ] リンクをクリックしてください。

プロジェクトの定義

フィルターなどのプロジェクト パラメーターを定義して、プロジェクトに関連付けられたデータを管理することができます。デフォルトでは、すべてのプロジェクトの全データに対してチームメンバーのアクセスが制限されます。ただし、プロジェクト マネージャーは必要に応じてプロジェクト フィルターを定義して、特定のプロジェクトへのアクセスをメンバーに許可できます。また、プロジェクトの定義で [制限なし] オプションが有効な場合、すべてのデータが利用可能になります。ただし、プロジェクト マネージャーはこのオプションを無効にしてデータに制限をかけることができます。

プロジェクト定義フィルターでのワイルドカードの使用

ワイルドカードとしてパーセント記号 (%) を使って、任意の文字または文字列を検索することができます。例を挙げて説明します。

たとえば、「7.1.1, 7.1.1.0, 7.1.1.23, 7.1.2.0」などのように C++test 7.1 のマイナーバージョンを個別に指定する代わりに、「7.1.%」と指定することができます。



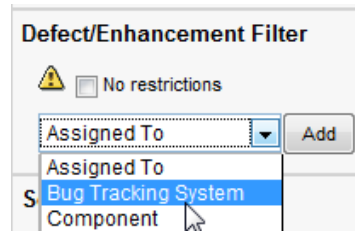
C++test 7.1 のすべてのマイルストーンが検索されます。

欠陥 / 改善フィルター

266 ページの「バグ トラッキング システムおよび要求管理システムの統合」で説明しているように、Project Center は数多くのバグ トラッキング システムと統合します。

プロジェクトとバグ トラッキング システムを関連付けるには、次の操作を行います。

1. [欠陥/改善フィルター] のドロップダウン リストから [バグ トラッキング システム] を選択して [追加] ボタンをクリックします。



Development Testing Platform (DTP) に統合されたバグ トラッキング システムの一覧が表示されます。このリストには、DTP の管理者が各システムに割り当てたラベルが使用されます。

2. 右側のシステムの一覧から、統合するバグ トラッキング システム (BTS) を選択し、[<] ボタンをクリックして BTS をプロジェクトと関連付けます。

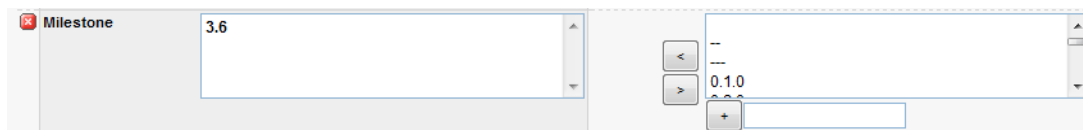


選択したバグ トラッキング システムのすべての項目がプロジェクトと統合されます。

特定のプロジェクトだけにフィルター条件を制限するには、次の操作を行います。

1. [欠陥/改善フィルター] のドロップダウン リストから [プロジェクト] を選択します。
2. [追加] ボタンをクリックします。これらは、統合されたバグ トラッキング システム からインポートされたプロジェクトです。
3. 右側のプロジェクトの一覧から、統合するコンポーネントを選択し、[<] ボタンをクリックしてこのコンポーネントをプロジェクトに関連付けます。

この方法で (つまり [欠陥/改善フィルター] のドロップダウン リストからフィルター カテゴリを選択し、左側のリストに目的のフィルター値を追加することで)、フィルターをさらに細かく設定できます。



ログ フィルター

ログ フィルターを使用すると、Report Center に送信されたログに基づいて制限を設定することができます。たとえば、Jtest から送信されたデータだけを表示するように Report Center を設定でき

まず、Report Center に結果を送信するツールはすべてログを作成します。ログはツールの実行についての記録です。ログには次の定義済みプロパティがあります。

- **マシン** ツールを実行したシステムがあるマシンの名前。
- **OS のアーキテクチャ** ツールを実行したシステムのアーキテクチャ。
- **OS の名前** ツールを実行したシステムがあるオペレーティング システムの名前。
- **OS のバージョン** ツールを実行したシステムがあるオペレーティング システムのバージョン。
- **ツール** テストを実行したツールの名前またはシンボル。たとえば Jtest 8.0 がテストを実行した場合、ツールは「Jtest」になります。
- **ツールのバージョン** テストを実行したツールのバージョン番号。たとえば Jtest 8.0 がテストを実行した場合、バージョンは「8.0」になります。
- **ユーザー** システムがツールを実行したときにログインしていたユーザーのアカウント。

上記のいずれかのログ フィルターに基づいて、データを制限するフィルターを作成するには、次の操作を行います。

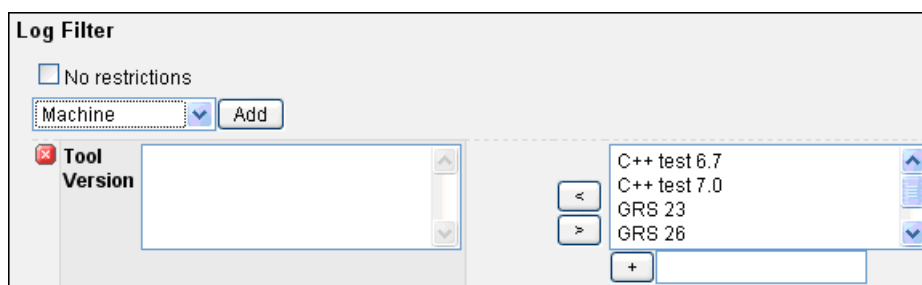
1. [ログ フィルター] で [制限なし] チェックボックスをオフにします。

プロジェクトフィルターの定義



2. [ログ フィルター] ドロップダウン リストから項目を選択し、その隣の [追加] ボタンをクリックします。

ログフィルター



3. 制限を指定します。

右側のリストには、すでに Report Center に送られたプロパティのすべての値が表示されます。一部の値はインストール時に Report Center データベースに追加されます。そのため、まだ一度もツールを実行していない場合でもツールの値がリストに表示されます。

- **ログによる制限を設定するには** 右側のリストから適切な項目を選択し、[<] ボタンをクリックします。選択した項目が左側のリストに移動します。左側のリストにある項目によってデータが制限されます。
- **制限を削除するには** 左側のリストから項目を選択し、[>] ボタンをクリックします。選択した項目が右側のリストに戻ります。
- **Report Center データベースにまだない値（したがってリストに表示されない値）を使って制限を設定するには** 右側のリストの下にあるテキスト フィールドに値の名前を入力し、[+] ボタンをクリックします。入力した値がリストに追加されます。

注意：ワイルドカード (%) を使って検索を拡張することができます。詳細については 180 ページの「プロジェクト定義フィルターでのワイルドカードの使用」を参照してください。

4. 構成を保存します。

ログ プロパティ フィルター

各ログは、固定のプロパティだけでなく属性を持つことができます。ログ プロパティ フィルターを使用すると、属性に基づいてデータをフィルタリングできます。

ログ プロパティ フィルターを設定するには、次の操作を行います。

1. [ログ プロパティ フィルター] の [制限なし] チェックボックスをオフにします。
2. ドロップダウン リストから適切なログ プロパティを選択し、[追加] ボタンをクリックします。
追加したいログ プロパティがない場合、次のステップに進んでください。

ログ プロパティ フィルター

The screenshot shows a window titled "Log Properties Filter". Inside, there is a text input field containing the word "Build", followed by a small downward-pointing arrow icon. To the right of this field is a button labeled "Add". Further to the right is another button labeled "Add New".

3. 次のいずれかのタスクを実行します。
 - **新しいログ プロパティをリストに追加するには：** [新規追加] ボタンの隣のフィールドにログ プロパティの名前を入力して [新規追加] ボタンをクリックします。入力したログ プロパティが表示されます。
 - **複数のログ プロパティを設定するには：** ログ プロパティは必要なだけ設定することができます。プロジェクトでは、1 つのログにつき 1 つの構成だけが存在することができます。
 - **ログ プロパティを削除するには：** 削除するログ プロパティの隣の赤い [+] ボタンをクリックします。
4. [プロジェクト フィルターの定義] タブの一番下の [保存] ボタンをクリックします。

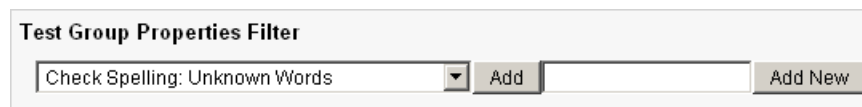
テスト グループ プロパティ フィルター

テスト グループ属性を利用してデータをフィルタリングすることができます。テスト グループ プロパティ フィルターは、ログ プロパティ フィルターと同じように動作します。異なる点は、テスト グループ プロパティ フィルターの属性はログではなくテスト グループに割り当てられることです。

Report Center は数多くのテストを処理します。テスト結果を分かりやすくまとめるために、テスト結果は「テスト グループ」に分類されます。テスト グループは一連のテストと他のテスト グループから構成されます。ログ フィルターの場合と同じように、各テスト グループは関連する属性を持ちます。テスト グループの場合、属性はテスト実行全体ではなく特定のテスト グループを表します。たとえば、静的解析テストのテスト グループは静的解析に関連する属性を持ちます。ホワイト ボックス テストのテスト グループはホワイト ボックス テストに関連する属性を持ちます。

テスト グループ プロパティ フィルターを利用すると、Report Center のレポート処理のためのプロジェクトをより柔軟に定義することができます。テスト グループ プロパティ フィルターの適用方法はログ プロパティ フィルターの場合と同じです。

テスト グループ プロパティ フィルター



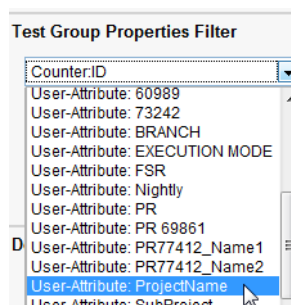
例

フィルターを作成して、DTP が受け取る自動テスト結果の中でどのテスト結果を特定のプロジェクトに関連付けるかを定義できます。

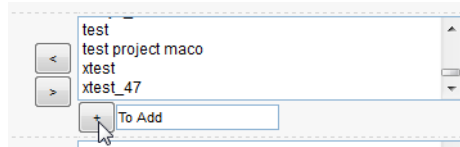
そのようなフィルターを定義する前に、Parasoft Test ツールが DTP に結果を必ず送信するようになる必要があります。詳細については 259 ページの「Parasoft Test から Report Center/Project Center へのテスト結果の送信」を参照してください。この設定が完了したら、Parasoft Test ツール (Jtest、C++test、dotTEST など) によって実行される自動テストは "User-Attribute: *ProjectName*" でマークされた結果を送信します。

関係のある自動テストをプロジェクトに関連付けるよう Project Center を構成するには、次の操作を行います。

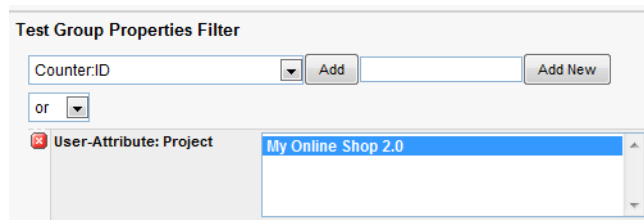
1. [テスト グループ プロパティ フィルター] ドロップダウン リストから [User-Attribute: *ProjectName*] を選択します。



2. [追加] をクリックします。以下のプロジェクトが表示されます。
 - a) 検索条件に一致している、および
 - b) DTP にテスト結果がある。
 - 目的のプロジェクト名がリストにない場合、そのような属性でマークされたテスト結果を DTP がまだ受け取っていないことを意味します。名前を入力して [+] ボタンを押して、そのような属性を手動で追加することができます。



3. 適切なプロジェクト名を選択し、[<] ボタンをクリックして左側のリストにプロジェクトを追加します。



ここまでの操作が完了すると、自動テストの結果が Parasoft Test ツールから DTP に送られて、ユーザー指定の属性（この例では "User-Attribute: ProjectKey=JBank1015"）でテスト結果がマークされている場合、そのテスト結果はプロジェクトに関連付けられます。

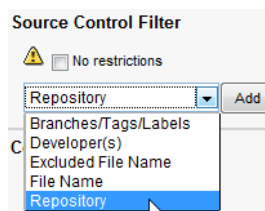
ソース管理フィルター

321 ページの「Report Center とソース管理システムの統合」で説明しているように、Project Center は複数のソース管理システムと統合します。

ソース管理システムと DTP を統合する場合、ソース管理システムは Parasoft Source Scanner によって定期的にスキャンされ、プロジェクトコードのリビジョン情報が Project Center データベースに渡されます。

Project Center とソース管理システムを統合するには、次の操作を行います。

1. [ソース管理フィルター] のドロップダウン リストから [リポジトリ] を選択します。



2. [追加] ボタンをクリックします。リポジトリの一覧 (DTP の管理者によって指定された Source Scanner プロジェクトの名前) が表示されます。

3. 右側の一覧から、統合するリポジトリの名前を選択して [<] ボタンをクリックし、プロジェクトのソース管理フィルターの一部としてこのリポジトリを追加します。

ソース管理フィルターをさらに制限するには、このように条件を追加していきます。たとえば、DTP プロジェクトに該当する特定のソースパスだけを指定するには、次の操作を行います。

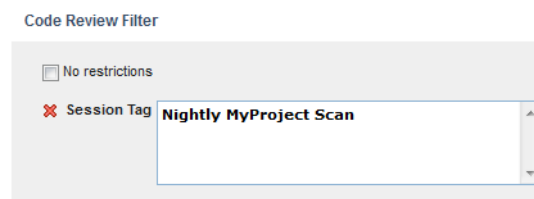
1. [ソース管理フィルター] のドロップダウン リストから [ファイル名] を選択します。
2. 値のフィールドで、関連するソースコードへのパスを指定します。
 - 指定したファイル名で開始するソースコードへのすべてのパスがプロジェクトに関連付けられます。
 - ワイルドカードとしてパーセント記号を使用できます。
3. [追加] ボタンをクリックします。ファイル名の一覧が表示されます。
4. ファイル名の一覧から、プロジェクトに関連付けるファイル名を選択します。
5. [<] ボタンをクリックして、プロジェクトに関連付けるファイル名としてこれを追加します。
6. [保存] ボタンをクリックしてプロジェクトの構成を保存して終了します。

既存のデータにこのフィルターを適用する場合、[再計算] ボタンをクリックします。

コード レビュー フィルター

コード レビュー フィルターは、Report Center のコード レビュー レポートに表示される結果をフィルタリングします。コード レビュー フィルターを定義する方法は、ログ フィルターの場合と似ています (181 ページの「ログ フィルター」を参照)。

コード レビュー フィルター



コード レビュー セッション タグはカスタム文字列です。コード レビューの結果に紐付けられ、コード レビュー データ パックを区別するために使用されます。このセッション タグは、Parasoft Test の [Code Review] テスト コンフィギュレーションのセッション タグと一致するように設定する必要があります (315 ページの「プロジェクトのコード レビュー レポートの設定」を参照)。

セッション タグを指定するほかに、次のようにフィルター制限をコントロールすることもできます。

- [制限なし] がオンの場合：
コード レビュー レポートには、選択されたプロジェクト チーム メンバーのすべてのコード レビューの結果が表示されます (どのプロジェクトやソース ファイルに対してコード レビューを実行したかは関係ありません)。
- [制限なし] がオフであり、セッション タグが定義されて設定されている場合：
DTP プロジェクトの設定で定義されたプロジェクト チーム メンバーが少なくとも一人コード レビューに関わっている限り、コード レビュー レポートには、選択されたセッション タグ

グでマークされたすべてのコード レビューの結果が表示されます。また、たとえ指定のセッション タグによる結果中にコード レビュー データが存在する場合でも、プロジェクト チームのリストにないユーザーのコード レビューは表示されません、詳細については 315 ページの「プロジェクトのコード レビュー レポートの設定」を参照してください。

Parasoft Test の設定

「現行プロジェクト」のテストを実行するために Parasoft Test 製品 (C++test、dotTEST、SOAtest、Jtest) を使用するとき、このページで設定した localsettings が使用されます。

Parasoft Test ツールごとにファイルを作成する代わりに、このページで一度に localsettings を設定できます。この設定は、Parasoft Test 製品が DTP プロジェクトに接続するとき、自動的にその Parasoft Test 製品に伝播されます。

例：

```
#Report Center Settings
grs.data.port=32323
#License Settings
license.network.host=ntp.company.com
license.network.port=2222
license.use_network=true
#Mail Settings
report.mail.domain=company.com
report.mail.from=john.doe
report.mail.password=123456789
report.mail.server=mail.company.com
report.mail.username=john
#Team Server Settings
tcm.server.accountLogin=true
tcm.server.enabled=true
tcm.server.name=ntp.company.com
tcm.server.password=123456789
tcm.server.port=18888
tcm.server.username=team_user
```

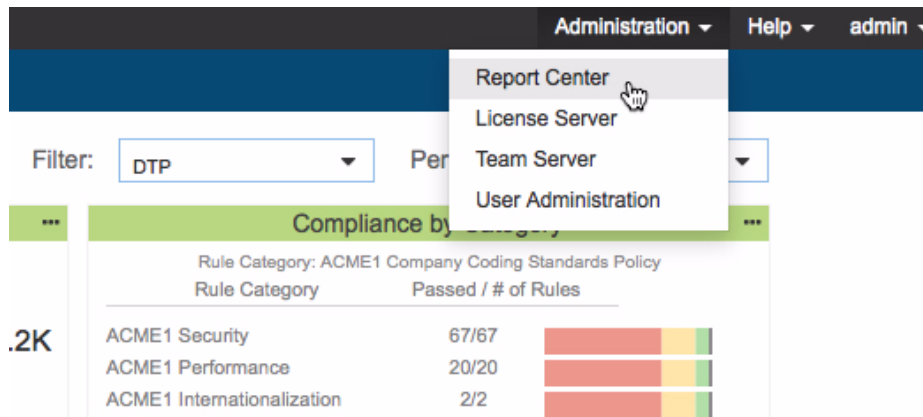
下記の値が定義されていない場合、DTP は自動的に値を設定します。Data Collector や Mail Server のポートなどのように、DTP の全般的な構成に基づいて値が設定されます。

```
#Report Center Settings
grs.data.port
#License Settings
license.network.host
license.network.port
#Mail Settings
report.mail.domain
report.mail.from
report.mail.password
report.mail.server
report.mail.username
#Team Server Settings
tcm.server.name
tcm.server.port
```

手動で値を設定すると、自動的に設定された値が上書きされます。詳細については『Parasoft Test ユーザーズ ガイド』を参照してください。

Report Center 管理ページ

管理者権限を持つユーザーは、Report Center 管理ページを使用できます。このページでは、プロジェクト、ツール、レポートについての管理設定を行うことができます。Report Center で [管理] ドロップダウンメニューから [Report Center] を選択すると、Report Center 管理ページが表示されます。



Report Center 管理ページには、Project Center 管理ページへのリンクと DTP ライセンス ページへのリンクがあります。詳細については以下の章を参照してください。

- 213 ページの「Project Center 管理ページ」
- 878 ページの「Parasoft License ページ」

Report Center 管理ページを終了するには、別の DTP ツールを選択します。



ユーザー権限の許可と管理については、603 ページの「[User Administration] ページ」を参照してください。

メインメニュー

[メイン] > [管理ホーム] を選択して、Report Center 管理ページに戻ります。

ラン コンフィギュレーション新規プロジェクトの追加

1. [プロジェクト] > [新規プロジェクトの追加] を選択します。
2. プロジェクトの名前および終了日と開始日を入力します。
3. [アクティブ] チェックボックスをオフにして、アクティブ化せずにプロジェクトを作成します。
4. [次へ] をクリックします。
5. 新規にチームを作成するか、既存のチームを選択します。新規にメンバーをプロジェクトに追加する方法については、180 ページの「プロジェクトの作成と構成」を参照してください。Report Center Administration で作成した新規ユーザーを追加すると、DTP にこの新規ユーザーが追加されます。
6. [保存] をクリックしてプロジェクト チームの画面を閉じます。
7. [完了] をクリックして新規プロジェクトを作成します。

Report Center は User Administration モジュールと密接にリンクしていて、User Administration モジュールは Report Center の多くの機能に影響します。たとえば、その 1 つは「チーム」の概念を導入するプロジェクト エディターです。「チーム」は（グループ メンバーと許可された権限に加えて）権限グループ構造の別名であり、User Administration モジュールで定義されます。チームを作成することによって、権限グループの構造全体が作成されます。User Administration モジュールの詳細については 609 ページの「権限グループについて」を参照してください。

既存プロジェクトの検索

1. [プロジェクト] > [プロジェクトの検索] を選択します。
2. [プロジェクト名] フィールドに名前を入力し、[検索] をクリックします。何も入力せずに [検索] をクリックすると、アクセスできるすべてのプロジェクトが表示されます。
3. プロジェクト名をクリックしてその詳細ページを表示します。

フィルターの作成と管理

フィルターとは、DTP に格納されたデータのカスタム ビューを可能にするラン コンフィギュレーションです (193 ページの「ラン コンフィギュレーションについて」を参照)。フィルターはデータベース中のプロジェクトに紐付けられ、Report Center ダッシュボードで使用することができます。さらに、REST API サービスの多くもパラメーターとしてフィルター ID を使用します。

デフォルト プロジェクト フィルター

DTP Engine からのレポートでプロジェクトが指定されていない場合、DTP Server はデフォルト プロジェクト フィルターに結果を割り当てます。以前のバージョンの DTP (5.1.1 以前) では、特

に指定しない限り、デフォルト プロジェクトはすべてのプロジェクトのデータを合計しました。DTP 5.1.2 からは、[新規ラン コンフィギュレーションを自動的に追加 (すべてのプロジェクトが対象)] オプションはデフォルトで無効です。これは、プロジェクトに割り当てられていない DTP Engine からのデータに対してフィルターを使用できるようにするためです。

ラン コンフィギュレーションについて

ラン コンフィギュレーションは、解析またはテストの「実行 (run)」を定義する属性セットです。ラン コンフィギュレーションは DTP Engine から DTP に送られます。解析を実行するエンジン、使用するテスト コンフィギュレーション、セッション タグ (ラン コンフィギュレーションで解析インスタンスを特定します)、および DTP のプロジェクト名によって一意に定義されます。同じプロジェクト名でタグ付けされた「実行」が、同じラン コンフィギュレーションにグループ化されます。

フィルターの作成

1. [プロジェクト] > [フィルター] を選択します
2. [新規作成] をクリックします。
3. フィルターの名前を入力し、このフィルターに関連付けるプロジェクトを選択します。

Filter Settings

The screenshot shows the 'Filter Settings' dialog box. The 'Filter Name' field is set to 'New Filter'. The 'Associated Project' dropdown menu is open, displaying a list of projects: Concerto, CopernicusImpl, CppTest, Cromwell (highlighted), Default Project, demo, Demo Project, dotTEST, dotTEST Ref Results, and DTP. Below the dropdown, there are checkboxes for 'Automatically add new' (checked) and 'Automatically add new' (unchecked). At the bottom, there are navigation buttons and a 'Test Configuration' button.


4. [ラン コンフィギュレーションの追加] をクリックして、DTP に送られたラン コンフィギュレーションを参照します。

5. [インクルード済み] 列で [追加] をクリックしてフィルターにラン コンフィギュレーションを追加します。

Add Run Configurations ✕

search filters

Run Configurations that are no longer able to have runs associated with them are marked with red. See documentation for more information.

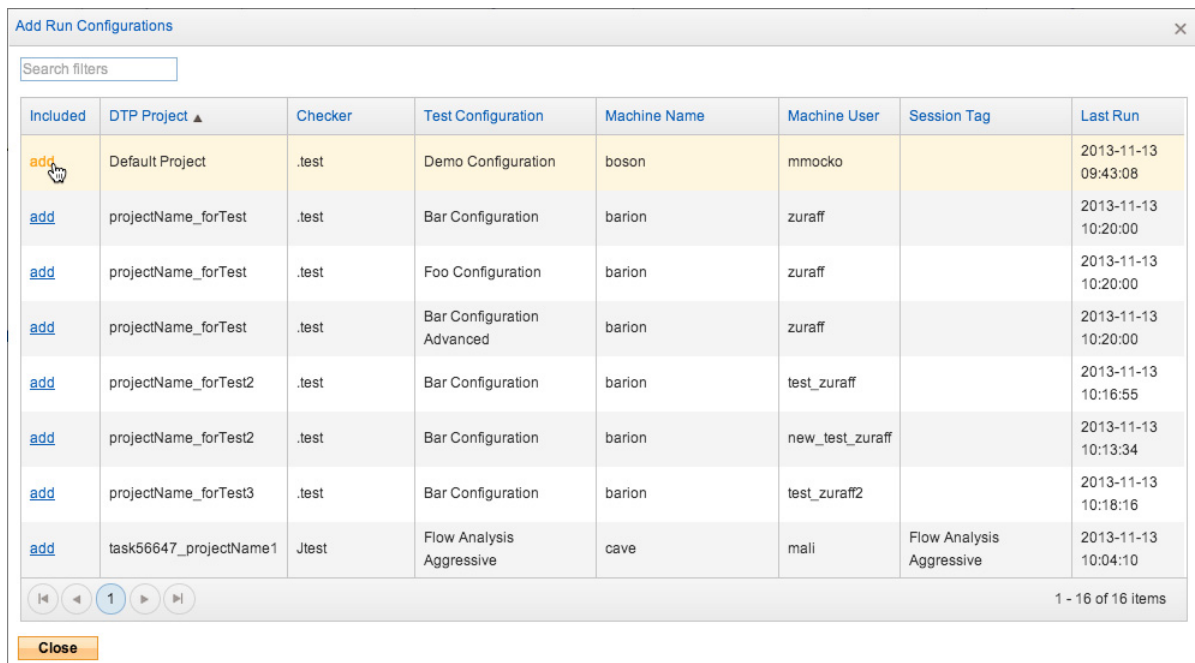
Included	Run Configuration Attributes				Last Run Information		
	DTP Project ▲	Engine	Test Configuration	Session Tag	Last Run	Last Machine Name	Last Machine User
	null	SOAtest	Demo Configuration		2014-12-02 19:49:08	soa-sdm-jtest.parasoft.com	jenkins
Add	null	SOAtest	Run Web Functional Tests in Firefox		2014-11-13 19:22:06	emqatester2	devtest
Add	null	SOAtest	DTP SOAtest - Linux x64 - MySQL 5.6		2015-01-12 19:43:23	snake.parasoft.com:18888	concerto-admin
Add	null	SOAtest	DTP SOAtest - Linux x64 - Oracle 11g		2015-01-09 22:49:54	snake.parasoft.com:18888	concerto-admin
Add	null	SOAtest	Example Configuration		2014-12-10 14:40:29	WKS2166L	baan4sc
Add	ATM	Parasoft C++test	Parasoft's Recommended Rules	Parasoft's Recommended Rules	2013-10-09 11:35:16	soa-sdm-jtest.parasoft.com	jenkins
Add	ATM	DTP Engine for C/C++	Recommended Rules	Recommended Rules	2015-07-23 10:38:16	cerberus	igarg
Add	ATM	DTP Engine for C/C++	Find Duplicated Code	Find Duplicated Code	2015-06-29 14:22:18	cerberus	igarg
Add	ATM	DTP Engine for C/C++	MISRA C 2012	MISRA C 2012	2015-07-23 10:38:45	cerberus	igarg
Add	Bank	DTP Engine for .NET	Find Duplicated Code	Find Duplicated Code	2015-06-29 07:47:38	cerberus	igarg

DTP 5.2.1 以降では、複数の別個の「実行 (run)」において、ラン コンフィギュレーション属性が同じであれば、実行する最後のマシン名あるいは最後のマシン ユーザー名が異なっても、それらの「実行」は同じラン コンフィギュレーションにグループ化されます。

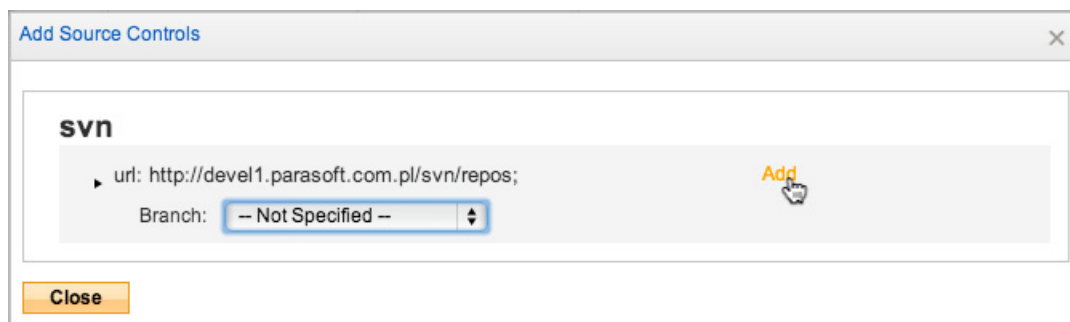
DTP 5.2.1 以降にアップグレードした場合、一部のラン コンフィギュレーションが「リタイア」となり、赤で強調表示されることがあります。その理由は、ラン コンフィギュレーション属性が同じで、実行する最後のマシン名あるいは最後のマシン ユーザー名が異なる場合に、5.2.1 より前の DTP では新規にラン コンフィギュレーションを作成したからです。

このように動作が変わったため、5.2.1 より前のラン コンフィギュレーションはその一意性が不明になり、したがって「リタイア」になります。

6. [閉じる] をクリックして [ラン コンフィギュレーションの追加] オーバーレイを閉じます。

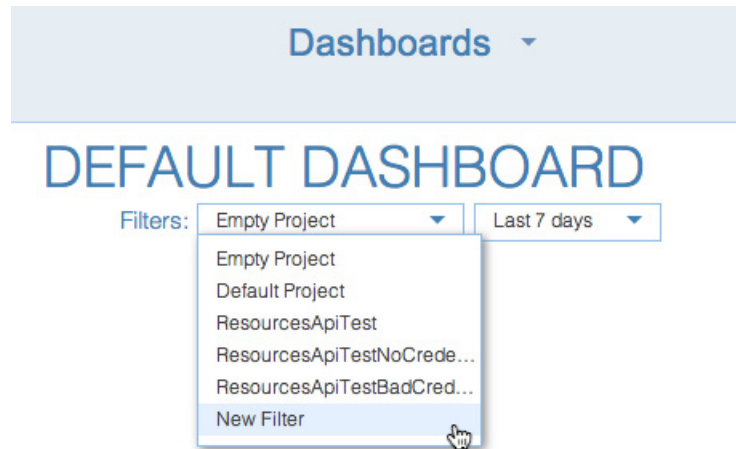


7. [ソース管理リポジトリ] をクリックして [ソース管理の追加] オーバーレイを開きます。このオーバーレイには、Parasoft Test 製品 (C++test、Jtest、dotTEST) から DTP に送られたソース管理のパスが含まれます。
8. ソース管理のブランチを参照し、[追加] をクリックしてフィルターに追加します。
9. [閉じる] をクリックします。



10. [含められた優先度] セクションに含める優先度を選択します。
11. [保存] をクリックして新規フィルターの作成を完了します。

新しいフィルターが Report Center ダッシュボードの [フィルター] ドロップダウン メニューに表示されます。



フィルターの編集

1. [プロジェクト] > [フィルター] を選択します。
2. フィルター名をクリックしてその設定を表示します。
3. 変更を加えて [保存] をクリックします。

フィルターの削除

1. [プロジェクト] > [フィルター] を選択します
2. フィルターの上にマウス ポインタを置き、[削除] をクリックします。
3. 削除を確認する問い合わせがあったら、削除を確認します。

フィルターの複製

1. [プロジェクト] > [フィルター] を選択します
2. フィルターの上にマウス ポインタを置き、[複製] をクリックします。
3. フィルターの名前を入力し、必要に応じて変更を加えます。
4. [保存] をクリックします。

カバレッジ イメージとフィルターの関連付け

カバレッジ イメージは、単体テストまたは機能テストの実行に関連付けられたカバレッジ データのための識別子です。Coverage Summary や Coverage Trend といったカバレッジ ウィジェットからの情報を表示するために、フィルターを正しいカバレッジ イメージに設定しなければなりません (102 ページの「テスト ウィジェット」を参照)。

カバレッジ イメージは、Code Coverage Engine (CCE) のプロパティ構成ファイルで指定され (詳細は DTP Engine のドキュメントを参照)、CCE が解析結果をレポートするときに自動的に DTP に送られます。カバレッジ イメージをフィルターに関連付けて、Report Center でカバレッジ情報を参照することができます。

1. [プロジェクト] > [フィルター] を選択します
2. フィルター名をクリックしてその設定を表示します。
3. ドロップダウン メニューからカバレッジ イメージを選択して [保存] をクリックします。

Filter Settings

GRS データベースの最適化

必要に応じて、古いデータや期限を過ぎたデータを手動で削除することができます。

1. [ツール] > [データベースの更新] > [データベースの整理 (GRS DB)] を選択します。
2. 必要に応じて日数を調整し、[実行] をクリックします。

指定した日数よりも古いデータがデータベースから削除されます。また、自動的にデータベースを最適化することもできます。207 ページの「データベースの自動最適化」を参照してください。

DB アップデーターのステータス

内部データベースのバージョン番号を含め、Report Center の現在のデータベース構造についての統計を表示するには、[ツール] > [データベースの更新] > [DB アップデーターのステータス] を選択します。

Data Collector へのレポートのアップロード

Parasoft コード解析ツールが生成した XML レポート ファイルを直接 DTP にアップロードすることができます。

1. [ツール] > [Data Collector アップロード フォーム] を選択します。
2. [ファイルの選択] をクリックし、アップロードするレポートを選択します。
3. [アップロード] をクリックします。
4. ブラウザーのナビゲーション ボタンを使って Report Center 管理ページに戻ります。

DTP に直接アップロードされたレポートのステータス確認

1. Report Center ダッシュボード ビューを開きます (36 ページの「ダッシュボードの使用」を参照)。
2. ダッシュボードに Data Collector Diagnostics ウィジェットを追加します (43 ページの「ウィジェットの追加」を参照)。

DTP が収集したデータについての情報がウィジェットに表示されます (69 ページの「分析結果ウィジェット」を参照)。

Data Collector Diagnostics				
Date X				
Date	Time ▼	Name	Status	
2014-03-18	11:34:49 AM	20140318-113449-report.xml	Processed	▲
▲ Date: 2014-02-20				
2014-02-20	1:26:14 PM	20140220-132614-_soa-sdm-jtest.parasoft.com_1392830488000 - copy.xml+recommended+rules+jenkins_	Processed	
▲ Date: 2014-01-16				
2014-01-16	10:03:02 AM	20140116-100302-_soa-sdm-jtest.parasoft.com_1389894801000.xml+owasp top 10+jenkins_	Processed	

ウィジェットがステータス エラーを返す場合は、レポート構成設定を確認してください。詳細については Parasoft Test または Static Analysis Engine のドキュメントを参照してください。

データベースのプロジェクトデータの再計算

次のコマンドは、プロジェクト ログ、テスト グループ、ソース管理エントリ、およびバグの数を再計算します。

1. [ツール] > [計算 (GRS DB)] > [プロジェクトの再計算] を選択します。
2. プロジェクト単位で再計算するには、プロジェクトの [実行] ボタンをクリックします。すべてのプロジェクトのデータを一度に再計算するには、一番下の [実行] ボタンをクリックします。

Recalculate Projects

Project	Logs	Test Groups	Source Control Entries	Bugs	Recalculate One Project
A	0	0	0	0	<input type="button" value="Execute"/>
A new	0	0	0	0	<input type="button" value="Execute"/>
Adamt1	0	0	0	4	<input type="button" value="Execute"/>
Default Project	NR	NR	NR	NR	<input type="button" value="Execute"/>

NR - not restricted by project.

Recalculate all projects:

GRS データベース中の成果物の再計算

欠陥と改善のテスト ステータスを再計算することができます。この計算を実行すると、更新されたステータスが Project Center の個々の欠陥と改善に表示されます。テスト ステータスは、Project Center > Defect/Enhancement Status レポートでも使用されます。

1. [ツール] > [計算 (GRS DB)] > [欠陥 / 要求の再計算] を選択します。
2. 次のいずれかのセクションで [実行] をクリックします。
 - **欠陥 / 改善のテスト ステータスのクイック計算**
前回の計算からの変更に基づいて欠陥 / 改善のテスト ステータスを計算します。
 - **欠陥 / 改善のステータスの完全計算**
欠陥 / 改善のテスト ステータスの完全な履歴を計算します。

実行ジョブの再計算

1. [ツール] > [計算 (GRS DB)] > [再計算ジョブの実行] を選択します。
2. [実行] をクリックします。

スキャンと計算の自動実行

デフォルトでは、BTS Scanner は 15 分ごとにインクリメンタル モードで実行するように設定されています。スキャンの直後に欠陥 / 改善のテスト ステータスのクイック計算が実行されます。詳細については 275 ページの「BTS Scanner の実行」を参照してください。

\$DTP_HOME/grs/config/CronConfig.xml を編集することで、ニーズに合わせて BTS Scanner のスキャンと計算の設定を変更することができます。次の表は、CronConfig.xml に追加できる設定についての説明です。

欠陥 / 改善の 計算オプション	CronConfig.xml の設定
バグ トラッキング システムからの欠陥 / 改善の完全な再スキャン	com.parasoft.grs.rserver.cronjobs.bts.BtsFullScannerJob
バグ トラッキング システムからの欠陥 / 改善のインクリメンタル再スキャン	com.parasoft.grs.rserver.cronjobs.bts.BtsScannerJob
欠陥 / 改善のテスト ステータスのクイック計算	com.parasoft.grs.rserver.cronjobs.RequirementCalculationJob
欠陥 / 改善のテスト ステータスの完全計算	com.parasoft.grs.rserver.cronjobs.RequirementFullCalculationJob

BTS Scanner は必要に応じて手動で実行することもできます。詳細については 275 ページの「BTS Scanner の手動実行」を参照してください。

Report Center データ キャッシュの消去

ユーザーが Report Center レポートに目を通すときに、データ キャッシュに格納されたすべてのデータを削除するには、次の操作を行います。

1. [ツール] > [無効化 (GRS DB)] > [データ キャッシュの無効化] を選択します。
2. [実行] をクリックします。

ファイル制限の無効化

ファイル制限は、Parasoft テスト ツールによってレポートされたファイルについての情報（たとえば失敗したコーディング規約テストなど）を、SourceScanner によってインポートされた特定の

ファイル情報にマッピングします。Report Center データベースに格納された現在のファイル制限を無効化するには、次の操作を行います。

1. [ツール] > [無効化 (GRS DB)] > [ファイル制限の無効化] を選択します。
2. [実行] をクリックします。

この処理には数分かかることがあります。

ログの削除

データベースから削除したいログを検索することができます。

- [ツール] > [ログの更新 (GRS DB)] > [ログの削除] を選択します。
- 検索条件を入力し、[Find] をクリックします (ワイルドカードはサポートされていません)。

Remove Logs

Find logs to remove.
Search using any field.

Machine name:

User name:

Tool name:

Date from (year/month/day): / /

Date to (year/month/day): / /

Log attributes:

Key:	Value:
1. <input type="text"/>	<input type="text"/>
2. <input type="text"/>	<input type="text"/>
3. <input type="text"/>	<input type="text"/>
4. <input type="text"/>	<input type="text"/>

Test group attributes:

Key:	Value:
1. <input type="text"/>	<input type="text"/>
2. <input type="text"/>	<input type="text"/>
3. <input type="text"/>	<input type="text"/>
4. <input type="text"/>	<input type="text"/>

3. 特定の行を削除するには、削除する行を選択して [削除] をクリックします。すべてのログを削除するには、すべてのログの削除用の [削除] をクリックします。

If you would like to delete all logs from search result, click here

Displaying 1-3/3 search results.

[1]

Log Id	Machine	User Name	Tool	Start Date	Remove
1	zangarous.parasoft.com.plthm		CVSScanner	2005-01-17	<input type="checkbox"/>
2	vivaldi2	User	Jtest	2005-01-17	<input type="checkbox"/>
3	vivaldi2	User	Jtest	2005-01-17	<input type="checkbox"/>

If you would like to delete logs selected on the current page, click here

いったん削除したログを復元することはできません。

ログの自動上書き

[重複ログの削除] は、Data Collector にとって不可欠な部分です。以前のログを自動的に上書きすることで、確実に最新のデータがデータベースに格納されます。ユーザーはメイン パラメーター（ツール、ユーザー、マシン）と属性キーを設定する必要があります。

1. [ツール] > [ログの更新 (GRS DB)] > [重複ログの削除] を選択します。
2. [削除の有効化] オプションを選択します。
3. パラメーター (203 ページの「[重複ログの削除] のパラメーター」) と属性キー (203 ページの「[重複ログの削除] の属性キー」) を入力します。
4. [保存] をクリックします。

Duplicate Logs Eradicator

Eradicator enabled

If Eradicator is enabled, all logs for specific tools, machines, users with the same values of attributes of top-level groups for all keys specified, will be overwritten with new ones. Tools, Machines and Users parameters are required. Use an asterisk (*) to indicate that any tool, machine, user are considered.

An asterisk (*) in 'Keys of attributes of top-level groups' field, means that all pairs (key,value) will be considered during group comparison.

Note: all overwritten logs will be removed from database irreversibly.

Apply Eradicator to:

Tools

*

Machines

*

Users

*

Keys of attributes of top-level groups

Keys

*

注意：いったん削除したログを復元することはできません。

[重複ログの削除] のパラメーター

以下のパラメーターを設定できます。

- **ツール**：ログの重複を防ぐために [重複ログの削除] が検証しなければならないツールを指定します。
- **マシン**：ログの重複を防ぐために [重複ログの削除] が検証しなければならないマシンを指定します。
- **ユーザー**：ログの重複を防ぐために [重複ログの削除] が検証しなければならないユーザーを指定します。

複数のツール、マシン、ユーザーを指定するには、項目をカンマで区切って指定します（たとえば Jtest, SOAtest, C++test）。すべてのツール、マシン、またはユーザーを指定するには、該当するフィールドにアスタリスク (*) を入力します。

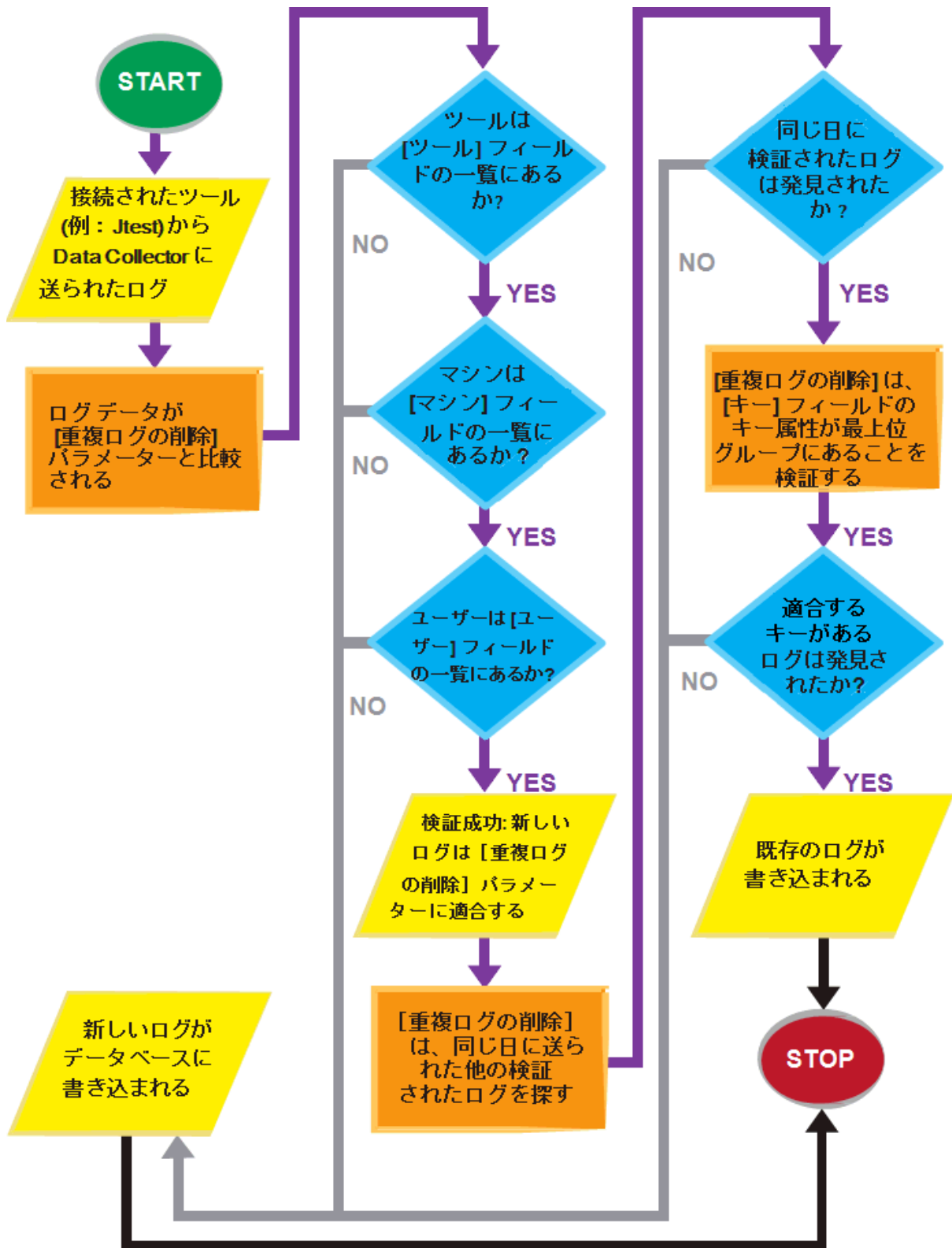
[重複ログの削除] の属性キー

最上位レベルのグループの属性キーを入力します。この属性キーを使って、[重複ログの削除] はログを比較し、上書きできるかどうかを判断します。

- 空白にした場合、新しいログ レポートが受け取られるときにテスト グループ キーは考慮されません。
- アスタリスク (*) を指定した場合、すべてのテスト グループ値が考慮されます。
- "User-Attribute: Project" は最もよく使用されるキーです。このキーは Parasoft Test GUI の General Project フィールドを参照します。[重複ログの削除] はこのキーを探し、General Project フィールドに同じ値がある結果について、ログを上書きします。
- カンマで区切って複数のキーを指定できます。たとえば：User-Attribute: Project, User-Attribute: TestMode
- 複数のキーを指定した場合、既存のログが削除されるには、指定したすべてのキーがテスト結果中に存在する必要があります。

重複ログの削除について

新しいログが Data Collector に送られると、[重複ログの削除] は設定されたパラメーターに対してログをチェックし、タイムスタンプを比較し、他のチェックを実行して、重複を防ぐためにログを上書きするかどうかを決定します。次の図は、[重複ログの削除] の検証プロセスを表したものです。



コード レビュー結果のスキャン

Team Server にある最新のコード レビュー スキャンの結果を強制的に Report Center に読み込むことができます。

1. [ツール] > [コード レビュー スキャン (GRS DB)] を選択します。
2. [実行] をクリックします。

CRHistoryHarvestingJob ジョブが実行されます。このジョブは、コード レビュー スキャンの結果を読み取り、Report Center データベースにデータを渡します。DTP_HOME/grs/config/CRHistoryScanners.xml ファイルで定義された設定を利用して Team Server に接続します。

データベース接続の管理

新規に Report Center データベース接続を作成したり、既存のデータベース設定を変更したりすることができます。GRS データベースと DTP データベースの接続を設定しなければなりません。どちらのデータベースも履歴データとテスト情報を格納しますが、DTP はダッシュボードとフィルター情報も格納します。詳細については 36 ページの「ダッシュボードの使用」、and 192 ページの「フィルターの作成と管理」を参照してください。

新規の接続を作成する

1. [設定] > [データベース] を選択します。
2. データベースの種類 (MySQL または Oracle) を選択します。
3. [新規作成] をクリックします。
4. 必須フィールドに入力します。
5. [作成] をクリックします。

既存の接続を変更する

詳細については 577 ページの「データベース接続の変更」を参照してください。

1. [設定] > [データベース] を選択します。
2. 変更を行います。
3. [保存] をクリックします。

電子メール通知の設定

チーム メンバーに電子メールで通知する設定の詳細については 541 ページの「タスク、欠陥、要求の電子メール通知」を参照してください。

1. [設定] > [E-mail] を選択します。
2. 電子メールの設定を入力します。

3. [保存] をクリックします。

Report Center データ キャッシュの有効化 / 無効化

データ キャッシュは、ユーザーがレポートを参照するときにデータベースから読み込まれるデータを格納します。

1. [設定] > [Report Center (GRS DB)] > [データ キャッシュ] を選択します。
2. [オン] または [オフ] を選択してデータ キャッシュを有効化または無効化します。
3. アイドル時間のフィールドに値を入力して、レポートを自動生成するまでレポート サーバーが待機する時間を設定します。
4. [保存] をクリックします。

Data Cache Settings

Switch Data Cache ON/OFF

ON

OFF

In ON mode GRS reports are internally auto-generated and put to Data Cache.
 Reports are auto-generated only between the hours specified in
 [PST]/grs/CronConfig.xml (Job id="Cache Report Executor")
 and after specific report server idle time is reached.
 The list of reports to be auto-generated is in [PST]/grs/staticLinksConfig.xml

Report Server idle time after which reports cache auto-generation starts

minutes

Note: When 0 minutes are set, reports cache auto-generation starts despite potential users browsing Report Server.

Save

データ キャッシュの詳細

ユーザーが Report Center レポートを生成するときに、DTP Server は次の情報をレポート統計に記録して保存します。

- 各レポートが (指定のパラメーターで) 表示された回数。
- 各レポートの生成にかかった時間。

レポート統計は、選択されたレポートを自動生成するために Report Center によって使用されます。レポート統計を参照して、Report Center は「最もよく使用されるデータ」と「生成にかなりの時間がかかるデータ」のいずれかをデータ キャッシュに保存します。そのようなデータはユーザーがすぐにアクセスできるよう、データ キャッシュに格納されます。データ キャッシュはユーザーの作業を著しくスピードアップします。しかし同時に、レポートが自動生成されるときに DTP Server を遅くすることがあります。管理者のタスクは、動作が遅く分断した状態ではなく迅速で順調になるよ

う、データ キャッシュの設定を構成することです。データ キャッシュの消去の詳細については、200 ページの「Report Center データ キャッシュの消去」を参照してください。

Data Collector の時間制限の設定

期間を設定して、データを Report Center データベースに直接保存するのではなく、一時的にディスクに保管することができます。Data Collector の時間制限を設定すると、データの取得と保存によって Report Center のパフォーマンスが大きく低下することがありません。

1. [設定] > [Report Center (GRS DB)] > [サーバー] を選択します。
2. [Data Collector の時間制限を有効化] オプションを有効にします。
3. 開始時刻と終了時刻を入力します。
4. [保存] をクリックします。

データベースの自動最適化

データベースからデータを自動的に削除するように Report Center を設定することができます。

1. [設定] > [Report Center (GRS DB)] > [サーバー] を選択します。
2. [データベースの最適化を有効にする] セクションで、Yes または No を選択してデータベースの自動最適化を有効または無効にします。
3. データを保管する最大の日数を入力します。
4. [保存] をクリックします。

非推奨の Team Server ベースのコード レビューの使用

Team Server ベースのコード レビューに切り替えることができます。その場合、Report Center のデータベースではなく Team Server からデータが取得されます。詳細については 315 ページの「コード レビューの統合」を参照してください。

1. [設定] > [Report Center (GRS DB)] > [サーバー] を選択します。
2. [非推奨の Team Server ベースのコード レビューを使用] オプションを有効にします。
3. [保存] をクリックします。

Policy Center マネージャーの E-mail 設定

どのチーム メンバーがマネージャーのレポートを受け取るかを設定します。マネージャー レポートは、すべてのプロジェクトを総合したレポートです。レポートするデータの範囲を設定したり、レポートを実行して特定ユーザーに送信する曜日を指定できます。

1. [設定] > [Policy Center] > [マネージャーの E-mail] を選択します。

2. 次のいずれかの操作を行います。
 - マネージャーのフィールドをクリックし、ドロップダウンメニューからユーザーを選択します。
 - [表示] をクリックし、オーバーレイでユーザーを選択し、[割り当て] をクリックしてマネージャー フィールドにユーザーを追加します。
 - [消去] をクリックしてマネージャー フィールドからユーザーを削除します。
3. レポート範囲を設定します。
4. マネージャー レポートを送信する曜日を選択します。
5. [保存] をクリックします。

License Server アクティビティ レポートの設定

License Server レポートに表示するデータを選択し、その表示方法をカスタマイズすることができます。関連するデータだけが分かりやすく表示されます。また、SOAtest や Jtest など、ツールに関連するデータをマージすることで、データの分離を減らし、詳細で明瞭なレポートを得ることができます。

1. [設定] > [License Server] を選択します。
2. ユーザー一人あたりの製品を数えるときにホストを考慮するかどうかを指定します。

License Server Activity - Reports Settings

Default Unique Users Recognition Policy:

username/host

- **Username/host:** 異なるホストの同じユーザーを、別のユーザーとしてカウントします。
 - **Username:** 異なるホストの同じユーザーを、一人のユーザーとしてカウントします。
3. [データのグループ化] ドロップダウンメニューからグループ化のタイミング（四半期単位または月単位）を選択します。

Group Data By:

month

4. レポートから除外する製品を選択します。

Do not show the following products on License Server reports:

000
100
1000

[Hold Ctrl button while clicking in order to select/deselect multiple items in the list]

5. [追加] をクリックして、マージされた共通レポートに追加する製品を選択します。あるいは、[削除] をクリックして製品を削除します。

Merge several products data into common product:

<div style="background-color: #007bff; color: white; padding: 2px;">C++Test</div> <div style="padding: 2px;">Jtest</div> <div style="padding: 2px;">SOAtest</div>	merge into common product name: <input style="width: 150px;" type="text"/>	<input style="background-color: #ffc107; border: none;" type="button" value="Delete"/>
Add new merge product names mapping	<input style="background-color: #ffc107; border: none;" type="button" value="Add"/>	

6. [Parasoft 問い合わせ先に四半期ごとのライセンス使用レポートを送信] オプションを有効にすると、四半期のライセンス使用データを自動的に収集し、問い合わせ先リストで指定された E-mail アドレスにレポートを送信することができます (211 ページの「Report Center 管理レポートの参照」を参照)。

7. [保存] をクリックします。

License Server アクティビティ レポートの詳細については 725 ページの「License Server の設定」を参照してください。

JMS イベント ブローカー ポートの設定

JMS イベント ブローカーは、DTP で発生するイベントについてリアルタイムな情報を提供します。デフォルトのポートは 61617 ですが、Report Center または Project Center の管理ページから変更することができます。

1. [設定] > [イベント ブローカーの構成] を選択し、[停止] をクリックします。
2. 新しいポート番号を入力し、[保存して再開] をクリックします。

現在サブスクリプションに使用できるトピックの説明を含め、JMS イベント ブローカーの使用の詳細については 853 ページの「DTP イベントのサブスクリプション」を参照してください。

開発コンポーネントとの統合

Report Center は、バグ トラッキング システムおよび要求管理システムと統合するほか、開発プロセスに可視性をもたらす他のサードパーティの開発ツールと統合します。バグ トラッキング システム、要求管理システム、および Java 対応 Emma コード カバレッジ ツールとの統合については、以下のセクションで説明します。

- Report Center と Emma の統合
- バグ トラッキング システムおよび要求管理システムの統合
- カスタム プロセッサとの統合

DTP の統合機能の詳細については 258 ページの「Development Testing Platform の統合」を参照してください。

Parasoft Test との接続

Parasoft Test は、Parasoft の開発テスト テクノロジーの構成、使用、相互運用を容易にするインフラストラクチャです。Parasoft Test を使用する Parasoft 製品（たとえば C++test、Jtest、dotTEST、SOAtest など）で定義されたプロジェクト設定をインポートするには、Parasoft Test と接続するように DTP を設定する必要があります。

1. [設定] > [Parasoft Test グローバル設定] を選択します。
2. Parasoft Test の設定を入力します（詳細については、対応する Parasoft 製品のドキュメントを参照してください）。
3. [保存] をクリックします。

ファイルエンコーディングについて

デフォルトでは、DTP はソースコードが UTF-8 でエンコーディングされていると想定します。異なる文字セットでソースコードをエンコーディングしている場合、`file.encoding.name` プロパティを追加して Parasoft Test の設定でそのエンコーディングを指定することができます。詳細については 187 ページの「Parasoft Test の設定」を参照してください。

テストコンフィギュレーションの管理

DTP Engines for Java、C/C++、.NET は、テストコンフィギュレーションにしたがってコードを解析します。テストコンフィギュレーションは、ルールや解析スコープのほか、ソフトウェア品質保証に関連する他のパラメーターを定義します。DTP Engine には、標準的なテストコンフィギュレーションが付属していますが、DTP からエンジンのユーザーにテストコンフィギュレーションを管理および配布して、開発テストポリシーを徹底させることもできます。

Report Center の管理メニューから [エンジン] > [テストコンフィギュレーション] を選択してテストコンフィギュレーションの管理ページを開きます。

テストコンフィギュレーションの編集と管理の詳細については、20 ページの「テストコンフィギュレーション」を参照してください。

Report Center サポート チームへのログの送信

DTP の使用で問題が発生した場合、サポート アrchive パッケージを作成して直接 Parasoft テクニカル サポートに送ることができます。

注意：以下の手順の送付先は米国の Parasoft サポートです。日本のテクニカル サポートの問い合わせ先については 722 ページの「テクニカル サポート」を参照してください

1. [設定] > [サポート] > [テクニカル サポート アrchive] を選択します。
2. 送信する情報を選択します。
3. [設定] ボタンをクリックして、Parasoft テクニカル サポートに連絡する E-mail を設定します。

4. [保存] ボタンをクリックして設定を保存します。
5. [作成] ボタンをクリックしてサポート アーカイブ パッケージを作成します。このパッケージは、DTP がインストールされているマシンの
DEVELOPMENT_TESTING_PLATFORM_HOME]/grs/grspackage.zip ディレクトリに保存
されます。

Report Center 管理レポートの参照

Report Center のアクティビティの把握に役立つ管理レポートが用意されています。

ログイン中のユーザー レポート

[レポート] > [ログイン中のユーザー] を選択すると、DTP に現在ログインしているユーザーが表示されます。開発チームの作業状況をモニタリングし、利用可能なライセンスが利用されていることを確認できます。

Users currently logged in		
User Login	Log In Time	Remote Host
grs	Tue Jul 05 16:10:57 PDT 2005	giotto.parasoft.com [10.10.24.35]
Users currently timed in		
User Login	User Time In	
adam	Tue Jul 05 13:01:56 PDT 2005	
dstover	Tue Jul 05 14:21:46 PDT 2005	
grs	Tue Jul 05 13:43:25 PDT 2005	

Data Collector アクティビティ レポート

データベースに格納するために Data Collector がデータを取得した時間、および同じデータベースから開発者がデータを取得した時間を確認できます。保管されるデータの差分といつそれが必要かを把握することで、データの収集と格納のプロセスを最適化できます。そうすることで、ボトルネックを防止し、データの移動を効率化することができます。

1. [レポート] > [Data Collector アクティビティ (GRS DB)] を選択します。
2. あらかじめ設定された期間を選択するか、カレンダー アイコンをクリックして期間を指定します。

3. [リフレッシュ] をクリックして、ページをリロードせずにレポートを更新します。

Last 30 days (SUM)													
h\min	05	10	15	20	25	30	35	40	45	50	55	60	Total
0	43	44	62	53	54	49	48	129	110	93	24	27	736
1	27	34	34	48	96	101	20	22	25	51	98	100	656
2	268	354	205	598	574	359	311	356	198	233	315	282	4053
3	200	335	230	209	257	171	255	297	163	84	67	83	2351
4	124	82	95	29	25	33	9	9	11	8	11	6	442
5	29	14	5	6	7	13	10	4	45	82	72	67	354
6	127	35	37	71	16	84	25	6	5	5	1	10	422
7	8	3		4	4	5	3	2	2	5	3		39
8	2	1	1		1	3	2	2	1	1		3	17
9	1	3	5	6	4	10	5	6	17	21	19	12	109
10	15	6	3	16	11	10	12	4	8	5	13	8	111
11	14	21	26	17	19	17	4	23	9	18	12	13	193
12	7	12	8	10	15	16	16	9	13	6	8	4	124
13	11	14	6	9	6	10	17	16	8	12	13	13	135
14	19	11	14	9	8	8	18	21	13	16	5	8	150
15	10	48	30	16	11	10	16	16	25	17	7	1	207
16	14	5	5	10	5	5	5	6	13	49	36	22	175
17	26	21	20	19	19	12	5	7	8	5	2	3	147
18	3	1	2	3	4	1		1	1		2	2	20
19	1	1	2	1		13	58	20	16	53	22	84	271
20	170	77	124	120	85	58	85	57	58	76	58	41	1009
21	445	424	223	184	212	176	80	82	97	97	94	74	2188
22	82	71	70	78	73	72	80	69	66	62	76	75	874
23	94	68	58	43	48	54	48	51	47	33	75	24	643
													15426

データベース ステータス レポート

[レポート] > [データベースのステータス (GRS DB)] を選択すると、Report Center データベースにあるすべてのテーブルとその行数が表示されます。Report Center の動作スピードが低下している場合、このレポートを参照してすべてのテーブルのサイズをチェックすることができます。

Database Statistics	
Table Name	# Rows
ACTIVITY	550752
BUG	49796
BUG_ACTIVITY	172923
BUG_ATTR	98950
BUG_ATTR_KEY	2
BUG_ATTR_TRIO	98950
BUG_ATTR_VALUE	349143
COD_FILE	98039
COD_LOG	11349
COD_MESSAGE	24629537

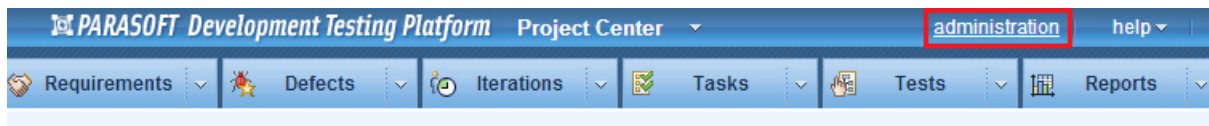
Project Center 管理ページ

管理者権限を持つユーザーは Project Center の管理ページにアクセスできます。タスクの通知、カスタムのステータスとフィールド、電子署名の要求を始め、さまざまな設定を行うことができます。このセクションでは、Project Center の管理ページで実行できる以下のタスクについて説明します。

- タスクと欠陥の通知メールの設定
- 成果物の名前変更
- カスタム ステータスの追加
- BTS Scanner と RMS Scanner の設定
- カスタム フィールドの使用
- 欠陥のプロパティの指定
- 電子署名と変更理由の設定
- Parasoft Test の設定
- SOAtest サーバーの設定

Project Center の管理ページを開く

1. 管理ページを開くには、Project Center で、ナビゲーションバーの [管理] リンクをクリックします。



2. 管理ページを終了するには、ナビゲーションバーの [Project Center] をクリックします。

タスクと欠陥の通知メールの設定

Project Center を設定して、タスクと欠陥の変更を電子メールで通知することができます。次の 2 種類の通知方法があります。

- **変更ベース**：タスクまたは欠陥が作成あるいは変更されると、電子メールが生成されます。このメールは所有者と監督者に送信されます。
- **日次サマリー**：その日のタスクの変更をまとめた 1 通のメールが毎日零時過ぎに生成されます。このメールは特定のプロジェクト リーダー（そのプロジェクトのマネージャーとして定義されているユーザー）に送信されます。

通知を設定する

1. [設定] メニューの [E-mail] を選択します。
2. [SMTP サーバー設定] で値を入力します。
 - サーバー（必須）

- ポート (必須)
- 接続のセキュリティ (必須)
- ユーザー名 (任意)
- パスワード (任意)
- Sender (任意)
- レルム (任意)

注意：" 任意 " のフィールドには、SMTP サーバーが必要とする場合にだけ値を入力してください。

E-mail Settings

SMTP Server Settings

Server:

Port:

Connection Security:

User Name:

Password:

Sender:

Realm: (Value is required only for those servers that authenticate using SASL realm)

[Test the settings](#)

E-mail address: (Test message will be sent to the specified e-mail)

E-mail address domain

User Default Domain: (When user has no email defined, the email is sent to : login@default_domain)

E-mail notification

Send e-mails when Project Center entities (e.g. tasks, defects) are changed, send daily digest e-mail to projects managers

3. [E-mail アドレス] フィールドに適切なメール アドレスを入力して [チェック] ボタンをクリックし、SMTP サーバーの設定をテストします。サーバーが正しく動作している場合、指定したメール アドレスにテスト メールが送信されて受信されます。
4. (任意) ユーザーのデフォルト ドメインのメール アドレスを入力します。
 タスクの変更についての通知は、User Administration モジュールで各ユーザーに設定されたアドレスに基づいてユーザーに送信されます (User Administration モジュールについては 603 ページの「[User Administration] ページ」を参照してください)。
 DTP でメール アドレスが定義されていないユーザーがいる場合、Project Center はこの管理ページで指定されたデフォルトのドメインアドレスをユーザーのログイン名に追加して、そのアドレスに通知メールを送信しようとします。
5. [Project Center タスクが変更されたときに E-mail を送信。プロジェクト マネージャーに日次のタスク サマリー E-mail を送信します。] チェックボックスをオンにします。
6. [保存] ボタンをクリックします。

成果物の名前変更

DTP では、「成果物」という用語は要求、欠陥、改善、反復、タスク、およびテストを指します。成果物のデフォルトの名前は、できる限り開発手法に中立なものになるように付けられています。定められた用語集や独自の名前付け規則がある場合、この成果物のデフォルト名を変更することができます。

1. [設定] メニューの [成果物の名前付け] を選択します。
2. 成果物に関連付ける新しい用語を指定します。次の図は、Scrum を採用しているチームの例です。

The screenshot shows the 'Artifact Naming Settings' page. At the top, there are navigation tabs for 'Main', 'Projects', and 'Settings'. Below the tabs, the page title is 'Artifact Naming Settings'. The main content area is divided into four sections, each representing a different artifact type:

- Defect:** Customized Name: Bug, Customized Short Name: Bug
- Enhancement:** Customized Name: Feature, Customized Short Name: Feature
- Iteration:** Customized Name: Sprint, Customized Short Name: Sprint
- Requirement:** Customized Name: User Story, Customized Short Name: US

3. [保存] をクリックします。

Project Center と Report Center の両方で、指定した名前がその成果物に割り当てられます

Project Center のデフォルトの成果物名に戻すには、[デフォルトに設定] をクリックします。

カスタム ステータスの追加

DTP では、欠陥、シナリオ、シナリオの実行、およびテスト セッションに対してチームが使用するカスタム ステータスを含めることができます。カスタム ステータスを使って、関連する成果物を分類、発見、およびレビューすることができます。

1. [設定] > [成果物のステータス] を選択します。
2. [新規] をクリックし、新しいステータスの名前を入力します。

Defect Statuses

5 Item(s) New

Active	Name	State
<input checked="" type="checkbox"/>	New	Unresolved - Editable
<input checked="" type="checkbox"/>	Assigned	Unresolved - Editable
<input checked="" type="checkbox"/>	Resolved	Resolved - Editable
<input checked="" type="checkbox"/>	Verified	Resolved - Read-only
<input checked="" type="checkbox"/>	New status	Unresolved - Editable

3. 成果物のステータスを指定します。編集可能ステータスを設定することで、成果物の他のプロパティも更新することができます（ランク、オーナー、期日など）。次の表は、成果物に設定できるステータスについての説明です。

成果物のステータス	利用できるステータス	定義	例
欠陥のステータス	未解決 - 編集可能	欠陥は未解決であり、そのプロパティは更新できます。	New
	未解決 - 読み取り専用	欠陥は未解決であり、そのプロパティは更新できません。	Assigned
	解決済み - 編集可能	欠陥は解決済みであり、そのプロパティは更新できます。	Resolved Fixed
	解決済み - 読み取り専用	欠陥は解決済みであり、そのプロパティは更新できません。	Verified Closed
要件のステータス	未解決 - 編集可能	要件は未解決であり、そのプロパティは更新できます。	Pending Review
	未解決 - 読み取り専用	要件は未解決であり、そのプロパティは更新できません。	Draft
	解決済み - 編集可能	要件は解決済みであり、そのプロパティは更新できます。	Pending Review
	解決済み - 読み取り専用	要件は解決済みであり、そのプロパティは更新できません。	Completed
シナリオのステータス	編集可能	シナリオは更新できます。	To Review
	読み取り専用	シナリオは更新できません。	Approved
シナリオ実行のステータス	編集可能	シナリオ実行は更新できます。	To Review
	読み取り専用	シナリオ実行は更新できません。	Approved

テストセッションのステータス	実行可能 - 編集可能	テストセッションは実行することができ、そのプロパティは更新できます。	Draft
	実行可能 - 読み取り専用	テストセッションは実行することができ、そのプロパティは更新できません。	Template
	実行不可能 - 編集可能	テストセッションは実行不可能であり、そのプロパティは更新できません。	Code Pending
	実行不可能 - 読み取り専用	テストセッションは実行不可能であり、そのプロパティは更新できません。	Legacy

4. このページの下部の [保存] ボタンをクリックします。

BTS Scanner と RMS Scanner の設定

DTP は、GUI を介して、または \$DTP_HOME/grs/config/bts ディレクトリの統合用 XML ファイルを設定することで、バグ トラッキング システムまたは要求管理システムと統合することができます。詳細については 266 ページの「バグ トラッキング システムおよび要求管理システムの統合」を参照してください。

欠陥のプロパティの指定

DTP の内部的な欠陥トラッキング システムを使用している場合、欠陥のオーナーと監督者のデフォルトを設定できます。

1. [設定] > [欠陥のプロパティ] を選択します。
2. [編集するフィールドの選択] ドロップダウン メニューをクリックして欠陥フィールドを選択します。

3. 値を入力して [新規] をクリックし、選択したフィールドに値を追加します。

Defect/Enhancement Fields Values Configuration

Project
Default Project

Choose field to edit
Component

Edited field: Component

1 Item(s) New Component Property New

Name	Owner	Observer(s)
Old Component Property		

カスタム フィールドの使用

DTP では、要求、欠陥、シナリオ、およびテスト セッションを分類するためにチームが使用するカスタム フィールドを含めることができます。ここで入力したカスタム フィールドは、関連する成果物を分類、発見、およびレビューするために使用できます。

カスタム フィールドを追加する

新しいカスタム フィールドを追加するには、次の操作を行います。

1. [設定] > [カスタム フィールド] を選択します。
2. ステータスを追加する成果物（たとえば欠陥、要求など）のタブで [新規] をクリックします。そしてテキスト フィールドの場合は [テキスト入力] を、コンボボックスの場合は [単一選択リスト] を選択します。


Custom Fields Settings

Requirement Defect Task Scenario Test Session

5 Item(s) New

Name	Field Type	Default Value	Built-in	Active	Unique Name
Parent	Text Input		true	true	tParent
Predecessor(s)	Text Input		true	true	tPredec
Type	Single Select List	Development	true	true	tType
task_custom1	Text Input		false	true	cf10
task_custom2	Text Input		false	true	cf11

3. カスタム フィールドの設定を指定します。
 - a. [ラベル] フィールドにフィールド名を入力します。
 - b. このフィールドについての説明を入力します。
 - c. フィールドをアクティブにするかどうか (ユーザーに見えるようにするかどうか) を指定します。
4. [保存] をクリックするか、次のステップに進んで単一選択リストのカスタム フィールドを設定します。
5. [利用可能な値] の表に値を入力し、[新規] をクリックします。このステップで入力した値はコンボボックスで使用できます。

 General

Label: *

Description:


Active:

Value Definition

Default Value:

Available Values

1 Item(s)

	Unique Name	Built-in	Name	
		true	--Not specified--	▲ ▼

6. 上下の矢印を使って、コンボボックスの値の順序を変更します。
7. 新しいカスタム フィールドのデフォルト値を入力します。
8. [保存] をクリックします。

フィールドをアクティブ化 / 非アクティブ化する

フィールドを非アクティブにすると、そのフィールドは UI から削除されます (フィールドそのものは削除されません)。管理ページではフィールドを再びアクティブにすることができます。

1. [設定] > [カスタム フィールド] を選択し、成果物のタブをクリックします。ページ下部の [アクティブなカスタム フィールドだけを表示] を有効にすると、非アクティブのフィールドを非表示にすることができます。


2. 編集アイコンをクリックします。

Main Projects Settings

Custom Fields Settings

Requirement Defect Task Scenario Test Session

1 Item(s) New

	Label	Field Type	Default Value	Built-in	Active	Unique Name
 Custom 1A Edit field definition		Text Input		false	true	cf4

3. [アクティブ] チェックボックスをオン/オフにして、UI に対してフィールドを追加 / 削除します。

General

Label:

Description:

Active:

Value Definition

Default Value:

欠陥のプロパティの設定









主要な欠陥フィールドで使用できる値を指定するには、欠陥のプロパティを設定します。たとえば、[検出バージョン] フィールドに値として 1.0、4.5、4.6、4.7 が表示されるように設定できます。

Choose field to edit

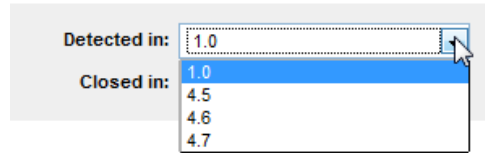
Detected in

Edited field: Detected in

4 Item(s) New

	Name	
	1.0	
	4.5	
	4.6	
	4.7	

ユーザーが欠陥を入力するとき、[検出バージョン] ドロップダウン リストが次のように表示されます。



電子署名と変更理由の設定

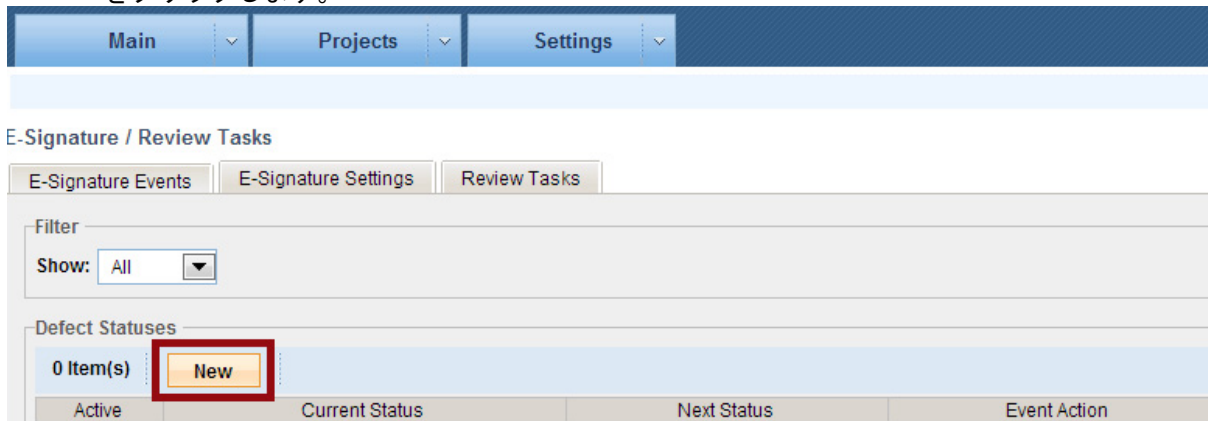
あるステータスから別のステータスに項目を変更するときに、必ず電子署名または変更理由を入力しなければいけないように Project Center を設定できます。たとえば、項目のステータスが "ドラフト" から "承認済み" に変わるときに電子署名を要求できます。入力された電子署名または変更理由は [変更履歴] タブに記録されます (392 ページの「電子署名 / 変更理由の入力とトラッキング」を参照)。

電子署名と変更理由の設定は、要求、欠陥、シナリオ、シナリオの実行、およびテスト セッションに対して指定できます。

電子署名または変更理由を追加する

新しい電子署名または変更理由を追加するには、次の操作を行います。

1. [管理] リンクをクリックして管理ページを開き、[設定] > [電子署名 / 通知の設定] を選択します。
2. 電子署名を追加する成果物 (たとえば欠陥、シナリオなど) のセクションで [新規] ボタンをクリックします。



3. [現在のステータス] および [次のステータス] で、いつ電子署名を要求するかを指定します。

E-Signature / Review Tasks

E-Signature Events | E-Signature Settings | Review Tasks

Filter
Show: All

Defect Statuses
0 Item(s) **New**

Active	Current Status	Next Status	Event Action
<input checked="" type="checkbox"/>	New	New	E-Signature

Requirement Statuses
0 Item(s) **New**

Active	Current Status	Next Status	Event Action
NO DATA TO DISPLAY			

4. [イベント アクション] フィールドで、電子署名を要求するか変更理由を要求するかを指定します。
5. [保存] ボタンをクリックします。

電子署名のメッセージを設定する

カスタムの電子署名メッセージを作成する方法については、393 ページの「カスタム電子署名」を参照してください。

電子署名の設定をアクティブ化 / 非アクティブ化する

電子署名の設定を非アクティブにすると、その電子署名は UI から削除されます。ただし、電子署名そのものは保持されるので、後で再び設定をアクティブにすることができます。

- [アクティブ] チェックボックスをオン / オフにして、UI に対して電子署名の設定を追加 / 削除します。
- ページ下部の [アクティブなカスタム フィールドだけを表示] を有効にすると、非アクティブの電子署名の設定を非表示にすることができます。

ステータス変更のレビュー タスクの設定

項目のステータスが変化したときにレビュー タスクを自動生成するように Project Center を設定できます。また、通知するユーザーグループや、自動生成されるレビュー タスクに割り当てるために使用するタスク作成ワークフローを指定することもできます。

たとえば、欠陥のステータスが "進行中" から "修正済み" に変わったときにレビュー タスクを作成するように DTP を設定することができます。さらに、タスク作成ワークフローをラウンド ロビンに設定することで、欠陥のステータスが初めて "進行中" から "修正済み" に変わったときに、レビュー タスクが生成されて QA チームのメンバーに割り当てることができます。次回、欠陥の変更ステータスが "進行中" から "修正済み" に変わると、レビュー タスクが生成されて別のチームメンバーに割り当てられます。この "ラウンドロビン" の割り当ては、チーム メンバー全員が 1 個のレビュー タスクを受け取るまで続き、その後ワークフローが繰り返されます。

DTP は、欠陥、シナリオ、シナリオ実行、およびテスト セッションのステータスが変化したときにレビュー タスクを作成するように設定することができます。

レビュー タスクを設定する

1. Project Center で、ナビゲーション バーの [管理] リンクをクリックし、[設定] > [電子署名 / 通知の設定] を選択します。
2. [レビュー タスク] タブをクリックし、[新規] をクリックします。レビュー タスクを自動生成するには、電子署名を定義する必要があります。
3. [イベント] 列で、いつレビュー タスクを生成するかを指定します。たとえば、成果物のステータスが "New" から "Verified" に変わったときにレビュー タスクを生成する場合、次の図のように設定します。

Defect Statuses		
Event	Review Task Owner - Member of Group	Create Tasks Workflow
New->Verified	Adamt team	Round Robin

4. [レビュータスクオーナー - グループのメンバー] からユーザー グループを指定します。このユーザー グループは、自動生成されたレビュー タスクを割り当てるチーム メンバーが属するグループです。このリストには、[User Administration] ページで指定されたすべてのユーザー グループが表示されます。
5. [タスク生成ワークフロー] フィールドで、ユーザー グループ メンバーにタスクを割り当てるために使用するワークフローを指定します。
 - Round Robin ワークフローでは、ステータスが変化したときに、グループ中の一人のメンバーに対してレビュー タスクが作成されます。
 - Notify All ワークフローでは、ステータスが変化したときに、グループ中の対応できるメンバー全員に対してレビュー タスクが作成されます。
6. [保存] ボタンをクリックします。

各項目タイプ (欠陥や要求など) のステータスが変化したときに、特定のチーム メンバーに対してレビュー タスクが自動的に作成されます。このレビュー タスクには "Review [artifact type] : [artifact ID] [artifact name]" の書式で名前が付けられます。[Description and Comments] の部分には「Status changed from [status 1]New to [status 1]. Click here to review: [artifact link].」という文章が自動的に挿入されます。

例 :

Project: *	JustTestProject
Name: *	Review Defect: [92,740] My defect here

Description and Comments

Description

Status changed from New to Assigned.

Click here to review: [PR 92740](#)

JMS イベント ブローカー ポートの設定

JMS イベント ブローカーは、DTP で発生したイベントについてリアルタイムの情報を提供します。デフォルト ポートは 61617 です。ただし、このポートは Report Center または Project Center の管理者ページから設定することができます。

1. [設定] > [イベント ブローカーの構成] を選択し、[停止] をクリックします。
2. 新しいポート番号を入力して [保存して再開] をクリックします。

現在サブスクリプションで利用できる機能を含め、JMS イベント ブローカーの詳細については 853 ページの「DTP イベントのサブスクリプション」を参照してください。

Parasoft Test の設定

「すべてのプロジェクト」のテストを実行するために Parasoft Test 製品 (C++test、dotTEST、SOAtest、Jtest) を使用するとき、このページで設定した localsettings が使用されます。「特定のプロジェクト」に対する localsettings の設定については 187 ページの「Parasoft Test の設定」を参照してください。

Parasoft Test ツールごとにファイルを作成する代わりに、このページで一度に localsettings を設定できます。この設定は、Parasoft Test 製品が DTP プロジェクトに接続するときに、自動的にその Parasoft Test 製品に伝播されます。

例：

```
#Report Center Settings
grs.data.port=32323
#License Settings
license.network.host=ntp.company.com
license.network.port=2222
license.use_network=true
#Mail Settings
report.mail.domain=company.com
```

```

report.mail.from=john.doe
report.mail.password=123456789
report.mail.server=mail.company.com
report.mail.username=john
#Team Server Settings
tcm.server.accountLogin=true
tcm.server.enabled=true
tcm.server.name=ntp.company.com
tcm.server.password=123456789
tcm.server.port=18888
tcm.server.username=team_user

```

下記の値が定義されていない場合、DTP は自動的に値を設定します。Data Collector や Mail Server のポートなどのように、DTP の全般的な構成に基づいて値が設定されます。

```

#Report Center Settings
grs.data.port
#License Settings
license.network.host
license.network.port
#Mail Settings
report.mail.domain
report.mail.from
report.mail.password
report.mail.server
report.mail.username
#Team Server Settings
tcm.server.name
tcm.server.port

```

自動的に設定された値は、ユーザーが手動で値を指定したときに上書きされます。

詳細については『Parasoft Test ユーザーズ ガイド』を参照してください。

SOAtest サーバーの設定

Project Center のインターフェイスから (SOAtest サーバーの Web サービスを介して) チームメンバーが SOAtest .tst ファイルを実行できるように設定できます。次の操作を行います。なお、現在 DTP は、アクセス制御が必要な場合、SOAtest サーバーへのアクセスをサポートしていません。DTP が SOAtest サーバーにアクセスする必要がある場合、アクセス制御が無効化された SOAtest サーバーを使用する必要があります。

1. [SOAtest Server の URL] で、SOAtest サーバーの Web サービス インターフェイスのサーバーとポート番号を入力します。

2. [SOAtest Test の設定] で、Project Center インターフェイスから SOAtest テストを実行するときにデフォルトで使用するテスト コンフィギュレーションを指定します。

SOAtest Server Settings

SOAtest Server Settings

SOAtest Server URL

Server:


Port:

SOAtest Test Settings


Default Test Configuration:

3. [保存] をクリックします。

これらの設定が SOAtest テスト タイプのデフォルトとして設定される点に注意してください。たとえば、あるテストで別のテスト コンフィギュレーションが必要な場合などには、必要に応じてこれらの設定をテスト レベルで変更できます。

 New Scenario

Scenario

 Scenario Steps

Step #1: Invoke SOAtest

SOAtest Server Parameters

Host:

Port:

SOAtest Test Suite Parameters

Path:

Configuration:

その他の Report Center 設定

このセクションでは、試験環境が実装された後に行う、必須ではないが推奨されるタスクについて説明します。

このセクションの内容：

- デフォルトの MySQL 設定の変更
- デフォルトの DTP Server ポートの変更
- Data Collector のデフォルト ポートの変更
- ユーザー アカウントの構成
- メトリクス データの保管期間の設定
- ソース コードのブランチ / タグと Report Center の連携
- ナビゲーション バーへの画像の追加

デフォルトの MySQL 設定の変更

デフォルトの Report Center/Project Center と MySQL の設定は、試験環境にとって非常に便利です。しかし、最終の実装では理想的ではありません。たとえば、mysql root パスワードはデフォルトでは空です。そのため、直接でもネットワーク経由でもサーバー マシンにアクセスできる人間であれば、顧客データをまったく間に破損することができます。状況は Report Center でも同じです。

次の表は、最終的なインストールで変更するべき設定のリストです。

アクションコード	対象	プラットフォーム	アクション	デフォルト設定	重要度
CONF-1	ntp オペレーティングシステム ユーザーアカウント	linux	パスワードの変更	"grs"	重要
CONF-2	mysql ルートアカウント	両方	パスワードの変更 アクセス権限の変更	パスワードなし	重要
CONF-3	mysql report center アカウント	両方	パスワードの変更 アクセス権限の変更	"report center"	重要
CONF-6	ntp 管理者アカウント	両方	パスワードの変更	"admin"	重要

CONF-1: DTP のユーザー アカウントを変更せずに、デフォルト パスワードを知っている権限のないユーザーがログインし、そのユーザー アカウントですべてのソフトウェアとバックアップを削除したり、その他のすべてを取り消したりすることができます。

CONF-2, CONF-3: mysql アカウントへのアクセスは、できる限り制限するべきです。理想的には mysql がインストールされているマシンにだけルート接続を許可することです。Report Center の接続は、Report Center ソフトウェア (DTP Server と Data Collector の両方) が実行中のマシンから可能であるべきです。なぜなら、これらのプロセスはどちらもユーザーが "grs" としてデータベースにアクセスすることを要求するからです。"grs" ユーザーは Report Center の Web インターフェイスからもアクセスできるべきです。マシンが定義されてなく、複数のマシンから接続を開始できる場合、セキュリティ制限はリモート マシンからの接続を制限するべきではありません。ローカルのサブネットからの接続を制限するべきです。すべてのケースにおいて、両方のアカウントのアクセス パスワードを変更するべきです。データベースへの接続を開放したままにしておくことは、権限のないアクセスやデータの破損を引き起こす可能性があります。

CONF-6: 管理者アカウントは、広く利用可能にするべきではありません。DTP の管理者は、レポートへのアクセス、ユーザーとグループの追加 / 削除、およびシステム管理ツールへのアクセスを許可したり取り消すことができます。

デフォルトの DTP Server ポートの変更

DTP のデフォルト ポートは 80 (Windows) または 8080 (Linux) です。ポートの変更は Team Server にも影響します。したがって、DTP Server に合わせて Team Server のポート設定も変更する必要があります。

UNIX の場合、DTP Server をポート 80 で起動するには root 権限が必要です。そのため、DTP Server の UNIX でのデフォルト ポートは 8080 に設定されています。

DTP Server の Web ポートをデフォルトから別のポートに変更するには、次の操作を行います。

1. DTP Server のサービスを停止します。
2. tomcat 構成ファイルを開きます。このファイルは次の場所にあります。

Windows の場合 : C:\Program Files\Parasoft\ntp\tomcat\conf\server.xml

UNIX の場合 : \$DTP_HOME/tomcat/conf/server.xml

3. デフォルトのポート番号の行に移動します。

```
<Service name="pst">
  <Connector port="80" maxThreads="150" minSpareThreads="25"
maxSpareThreads="75"
```

4. デフォルトの 80 を適切なポート番号に変更します。
5. server.xml ファイルを保存します。
6. Report Server を再起動します。

Data Collector のデフォルト ポートの変更

Data Collector のポートはデフォルトで 32323 に設定されています。Data Collector はこのポートを使って Parasoft ツールからすべての Report Center レポートのためのデータを取得します。Data Collector がインストールされているマシンのポートを変更する必要がある場合、次の操作を行います。

1. Data Collector サービスを停止します。
2. dtp\grs\config\DCCConfig.xml ファイルを編集します。

```
<data-collector-config>
    <!-- Changing data-collector-port parameter requires DataCollector
to be restarted -->
    <data-collector-port>32323</data-collector-port>
```

3. 適切なポート番号を入力します。
4. DCCConfig.xml ファイルを保存します。
5. Data Collector サービスを起動します。

ユーザー アカウントの構成

DTP にはあらかじめ定義されたアカウントがあります。admin、manager、developer、および architect と、同じ名前の定義済みグループです。ただし、運用環境では顧客のニーズを調べて別のアカウントとグループを作成することを推奨します。

アクション	例	構成の提案
CONF2-1	Report Center を使用する開発者のグループは、制限なしですべてのデータにアクセスできる。管理者アカウントだけが制限される。	"public" アカウントとグループを作成します。目的のレポートへのアクセスをこのアカウントに許可します。このアカウントのパスワードを全員に公開します。 "developer"、"manager"、および "architect" のアカウントを削除します。詳細については 692 ページの「名前付きアカウント (Named Accounts) の定義」を参照してください。

CONF2-2	Report Center を使用するユーザーに、区別できる役割を与えます。各ユーザーは、開発者、アーキテクト、またはプロジェクト マネージャーに分類できます。	ユーザーごとにアカウントを作成します。あらかじめ定義されたグループ "architect"、"developer"、および "manager" を使って、グループ メンバーごとに利用できるレポートとビューを区別します。顧客のニーズに合わせて、デフォルト アカウントを削除することも、単にブロックすることもできます。管理者アカウントの "admin" は残しますが、必ず 新しい安全なパスワードを用意する必要があります。詳細については 692 ページの「名前付きアカウント (Named Accounts) の定義」を参照してください。
CONF2-3	Report Center を使用するユーザーの人数が多く、開発者、アーキテクト、プロジェクト マネージャー、その他から構成される。これらのユーザーがたくさんのプロジェクトで作業している。	複数のプロジェクトからアクセスが必要なデータがない限り、役割 / プロジェクトごとにグループを作成します。たとえば、すべてのプロジェクトのデータをマネージャー達が参照できる場合、マネージャー承認に対して 1 つのグループだけを作成できます。ユーザーごとに個別のアカウントを作成し、アカウントを適切なグループに割り当てます。顧客のニーズに合わせて、デフォルト アカウントを削除することも単にブロックすることもできます。管理者権限を持つアカウントを複数作成しても良いでしょう。 アカウントの構成の詳細については 692 ページの「名前付きアカウント (Named Accounts) の定義」を参照してください。

メトリクス データの保管期間の設定

DTP Engine からのメトリクス データを DTP データベースに保管する日数を設定できます。

1. Data Collector サービスを停止します。
2. DTP_HOME/grs/config/DCCconfig.xml 構成ファイルを開き、以下の設定を探します。

```
<!-- Number of days that metric details should be kept in database. -->
<!--metric-details-retention-period>8</metric-details-retention-period-->
```

3. 設定のコメントを外し、データベースにメトリクス データを保管する日数を指定します。デフォルトは 8 日間です。

```
<metric-details-retention-period>16</metric-details-retention-period>
```

設定がコメントアウトされている場合、または値が指定されていない場合には、デフォルト値が使用されます。

4. 構成ファイルを保存し、Data Collector を再起動します。

ソースコードのブランチ/タグと Report Center の連携

Report Center の一部のレポートは、開発プロジェクトのソース ファイルに関する統計を表示します。ユーザーは、データの取得のためにスキャンされるソース リポジトリ ファイルを定義できるほか、そのフォルダー、命名パターンなどを定義することができます。

たとえば、あるソース リポジトリ ブランチでプロジェクトが開発される場合、または特定のタグのソース ファイルの統計を参照したい場合、ソース リポジトリのブランチおよびタグと連携するよう Report Center を構成できます。

Report Center とブランチ / タグと連携させるには、次の作業を行います。

- 特定のブランチ / タグをスキャンするよう SourceScanner を構成する
- Report Center プロジェクトに特定のブランチ / タグを割り当てる
- Report Center レポートでソースの統計を参照する

特定のブランチ/タグをスキャンするよう SourceScanner を構成する

SourceScanner は、指定された CVS ファイル (ブランチ / タグ) をスキャンしてデータを Report Center に送信します。必要に応じて Report Center でソース コードの変更を参照できるようにするには、たとえば `my_project_6_0_branch` など、適切なブランチまたはタグをスキャンするよう SourceScanner を構成する必要があります。

このタスクを完了するための詳細については『Parasoft SDLC Integration Extensions User's Guide』を参照してください。

重要！ 1 つの特定のソース リポジトリの場所をスキャンするのは、1 つの SourceScanner プロジェクトだけにすることを強く推奨します。たとえば、同じファイルの 2 つのブランチとトランクをスキャンする場合、これは 1 つの SourceScanner プロジェクトで定義するべきです。3 個の異なるプロジェクトを定義して実行してはいけません。

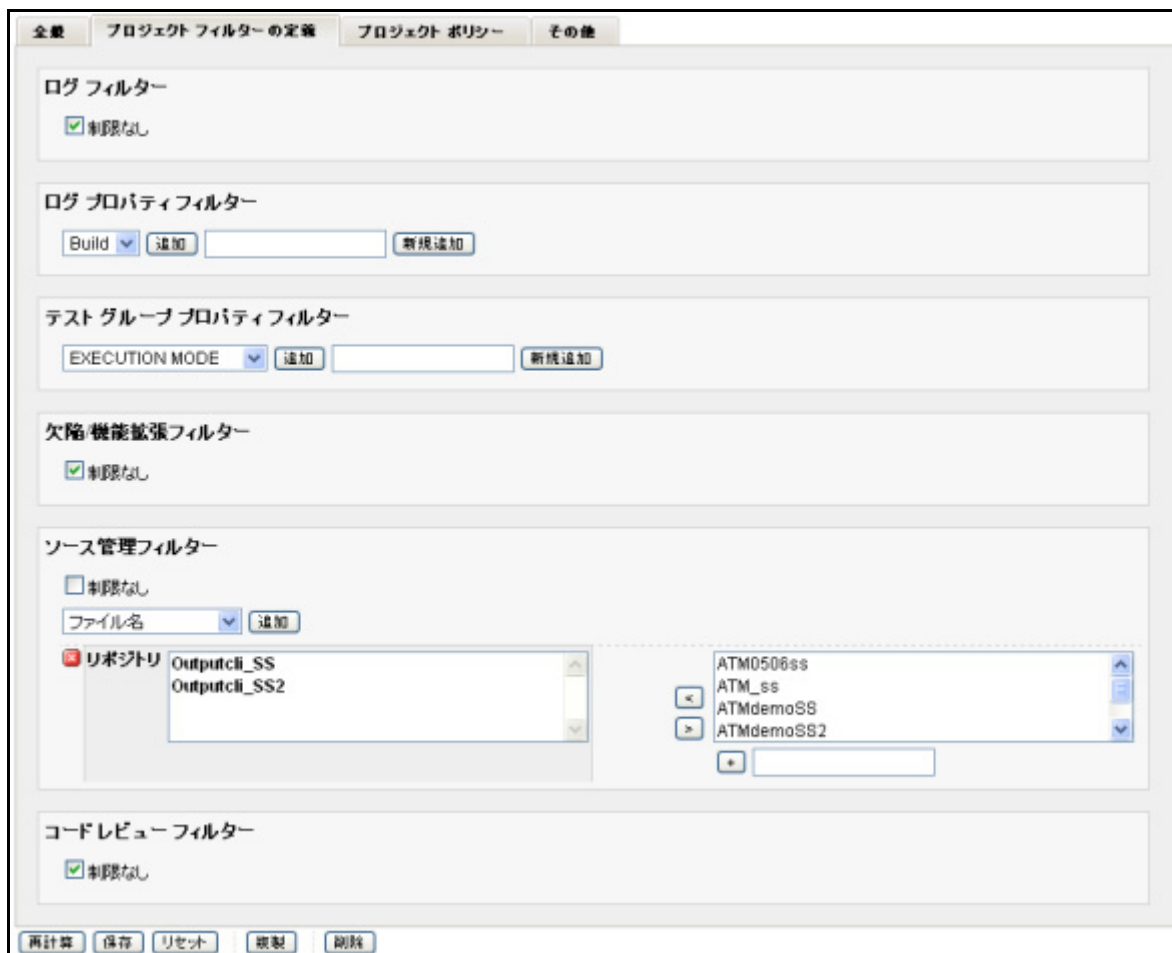
Report Center プロジェクトに特定のブランチ/タグを割り当てる

Report Center プロジェクトを構成したら、SourceScanner がスキャンするソース ファイルの中から、どのファイルを指定の Report Center プロジェクトに表示するかを指定する必要があります。

1. Report Center の [メイン] メニューの [管理] をクリックして管理モードに移行し、[プロジェクト] メニューの [検索] をクリックします。

適切なプロジェクトを探すか、新しいプロジェクトを作成します (192 ページの「ラン コンフィギュレーション新規プロジェクトの追加」を参照)。

2. [プロジェクト フィルターの定義] タブをクリックし、[ソース管理フィルター] セクションを表示します。



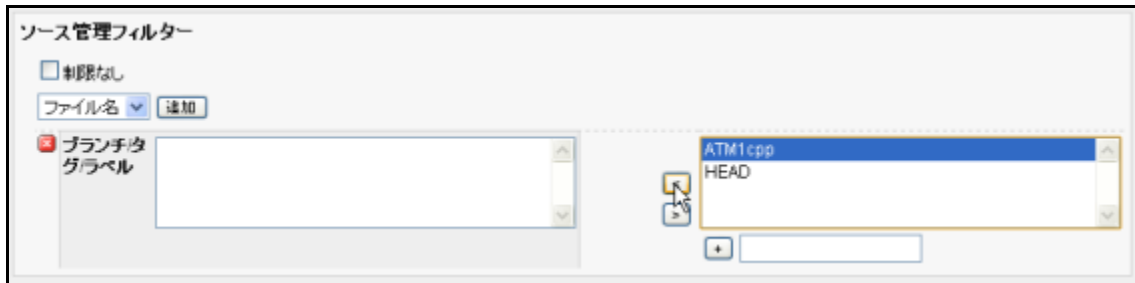
3. [ソース管理フィルター] セクションで制限を指定します。

- a. ドロップダウン リストから [ブランチ / タグ / ラベル] を選択して [追加] ボタンをクリックします。



右側のウィンドウには、これまでに SourceScanner がスキャンしたすべてのブランチ / タグの名前が表示されます。

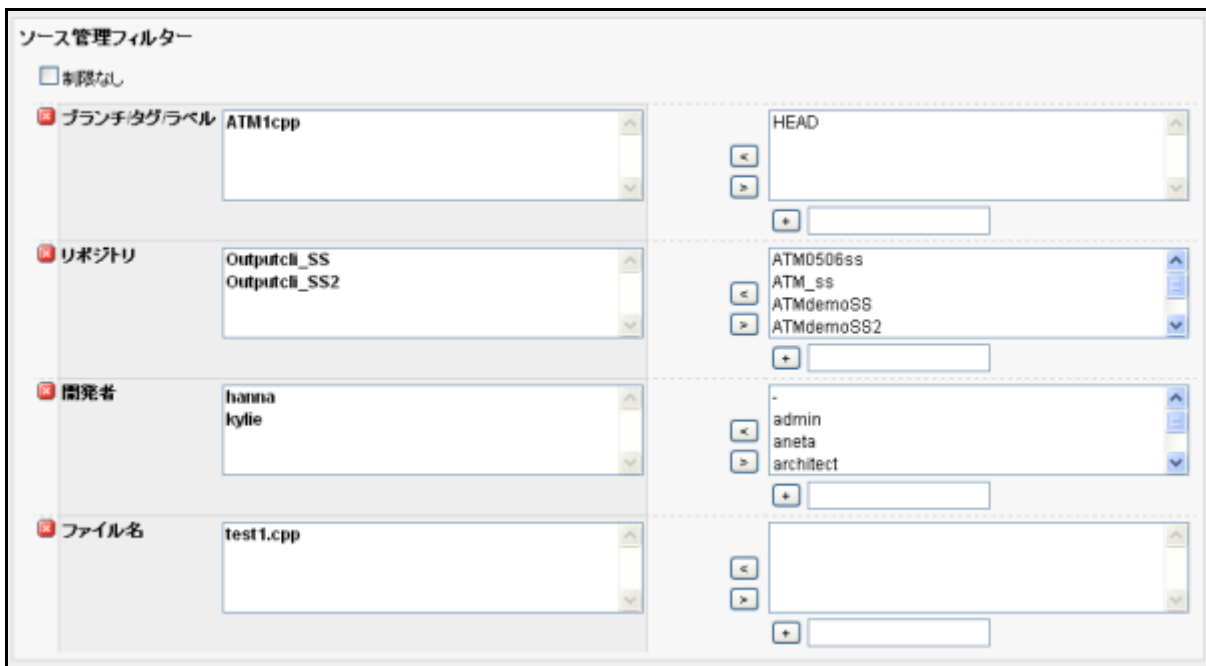
- b. 適切なブランチ / タグ / ラベルを選択し、[<] ボタンをクリックします。



選択したブランチがこの Report Center プロジェクトの [ソース管理フィルター] の一部としてマークされます。

- c. (オプション) [開発者]、[ファイル名]、または [リポジトリ] について制限を設定する必要がある場合、ステップ a および ステップ b に似た操作を行います。

注意: 通常、開発者、ファイル名、およびリポジトリの制限を設定するのは、プロジェクトの [ソース管理フィルター] を定義するときです。



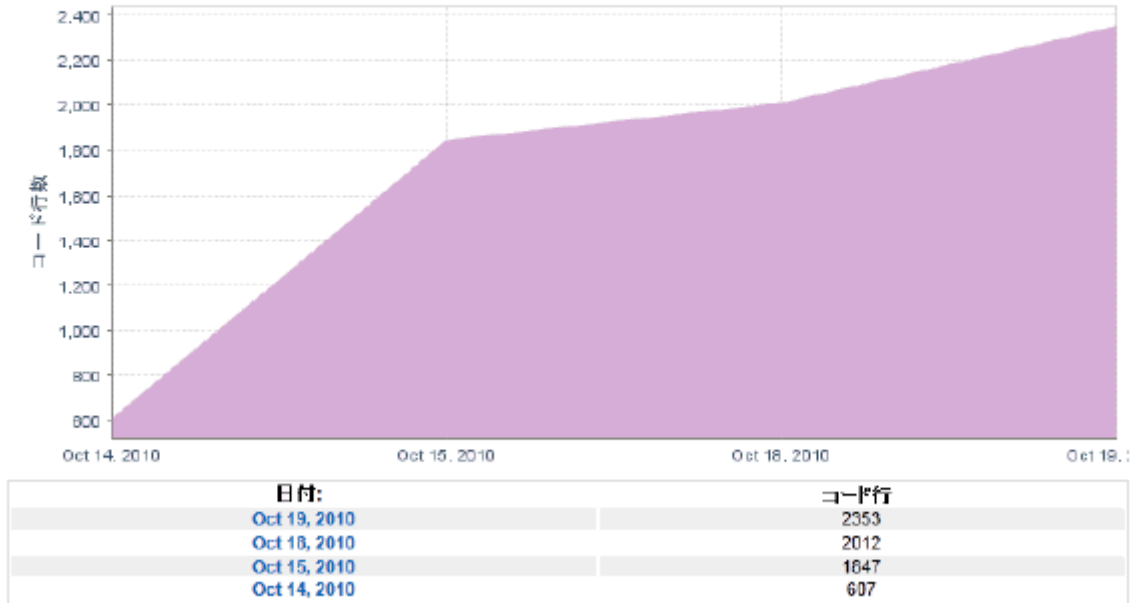
4. 構成を保存し、プロジェクトを再計算します。

Report Center レポートでソースの統計を参照する

SourceScanner と Report Center で構成したソース ブランチの統計を参照するには、ソース ファイル データを表示する任意の Report Center レポートに移動します。そのようなレポートとして、たとえば Base Size レポート (下記の図) と Check-ins レポートがあります。

プロジェクト PolicyDemo 任意の値 最新 7 | 15 | 30 日 最新 12 | 26 | 52 週 最新 10 | 2

コードベースのサイズ



[\[プロジェクトソースコードを表示\]](#)

ページの上の方に [Restrict Code - From Branching Point Only] というオプションがあります。

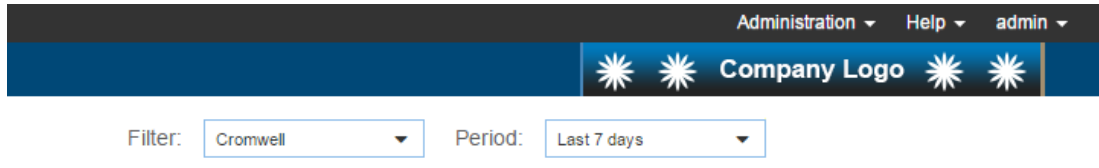
- [Restrict Code - From Branching Point Only] が有効な場合、ブランチングポイントが発生した日の後にコミットされたファイルリビジョンだけが、レポートの計算の対象となります。

ブランチングポイントは、リビジョン中でブランチが作成されたときに発生します。言い換えると、ブランチがメイントランク/ヘッドまたはその親ブランチから分離されます。

- [Restrict Code - From Branching Point Only] が無効な場合、すべてのファイルリビジョンがレポートの計算の対象となります。

ナビゲーションバーへの画像の追加

ナビゲーションバーには画像を追加することができます。DTP に企業やチームのロゴを表示することができます。



サポートされるファイル形式は PNG だけです。ファイル名は custom-logo.png でなければなりません。画像は横 300 ピクセル、縦 35 ピクセルに拡大 / 縮小されます。

1. 既存のプレースホルダー（空白）のカスタムロゴを削除します。
[DTP_HOME]/tomcat/webapps/grs/assets/custom-logo.png
2. ステップ 1 と同じ場所に新しい PNG ファイルを追加します。ファイル名は custom-logo.png でなければなりません。

Cache Report Executor の設定

Cache Report Executor のジョブは、staticLinksConfig.xml 構成ファイルで定義されたレポートを参照し、キャッシュします。デフォルトでは、このジョブは午前 5 時 (allowFromHour=5) に開始するように設定されています。デフォルト設定がニーズに合わない場合、Cache Report Executor ジョブを変更して、キャッシュ開始時刻とキャッシュ対象のレポート リストを変更することができます。

Cache Report Executor について問題があった場合は、353 ページの「Cache Report Executor」を参照してください。

キャッシュ開始時刻の設定

キャッシュ開始時刻のスケジュールは、CronConfig.xml ファイルの Job id="Cache Report Executor" xml 要素で設定することができます。

キャッシュするレポート リストの設定

Cache Report Executor のジョブが実行されたときにキャッシュするレポートのリストを設定するには、staticLinksConfig.xml ファイルを編集します。次はこのファイルの例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<reports>
  <report descr="Architect Dashboard" xreport-id="architect_dashboard/
composite_desc" composite="true" params=""/>
  <report descr="Practices > Coding Standards" xreport-id="practices/
cs_composite_desc" composite="true" params=""/>
  <report descr="Practices > White Box" xreport-id="practices/wb-bb/
wb_composite_desc" composite="true" params=""/>
  <report descr="Practices > White Box > Coverage" xreport-id="practices/
wb-bb/WBCoverageDetails" composite="false"
params="period=10drops&dateMode=period"/>
  <report descr="Practices > Black Box" xreport-id="practices/wb-bb/
bb_composite_desc" composite="true" params=""/>
  <report descr="Practices > Black Box > Coverage" xreport-id="practices/
wb-bb/BBCoverageDetails" composite="false"
params="period=10drops&dateMode=period"/>
  <report descr="Audit > Drop Grade" xreport-id="audit/DropGrade"
composite="false" params="period=10drops&dateMode=period"/>
  <report descr="Audit > Errors By Category" xreport-id="errors/
ErrorsByCategory" composite="false"
params="period=10drops&dateMode=period"/>
  <report descr="Audit > Errors By Severity" xreport-id="errors/
ErrorsBySeverity" composite="false"
params="period=10drops&dateMode=period"/>
</reports>
```

次の表は、上記のファイルに含まれるタブの一部とその説明です。

タグ	説明
reports	ルート要素。
report	キャッシュする 1 つのレポートについての情報を格納します。
descr	キャッシュするレポートについての簡単な説明。
xreport-id	キャッシュするレポートを特定する xreport 識別子を格納します。
composite	レポートが複合レポートの場合、この属性を true に設定します。そうでない場合、false に設定します。
params	レポートをキャッシュするために使用されるレポート パラメータを持つ属性。アンパサンド (&) を使って複数のパラメータを連結します。

キャッシュするレポートの追加

xreport-id 属性は、適切なレポートの URL をコピーして設定することができます。例としてアーキテクト ダッシュボードを使って説明します。

1. Report Center の Web ページに移動します。

```
http://localhost/grs
```

2. キャッシュするレポートに移動します。たとえば：

```
http://localhost/grs/
xarchitect_dashboard.jsp?xreportcomposite=architect_dashboard/
composite_desc.
```

3. リンクをコピーします。この例では architect_dashboard/composite_desc です。これを xreport-id 属性の値として貼り付けます。
4. composite 属性が true であることを確認します。

Report Center とレポート キャッシュのメカニズム

Report Center のレポートは次の 2 つの方法でキャッシュされます。

- ユーザーがブラウズしてレポートを生成したとき、レポートはキャッシュに格納されます。
- 毎晩 Report Center のバックグラウンド ジョブが実行するときに、`ntp\grs\config\staticLinksConfig.xml` 構成ファイルで指定されたレポートがキャッシュされます。

レポート キャッシュは次の 3 つの方法で無効化できます。

- Report Center バックグラウンド ジョブの夜間 (夜の 12 時以降) の実行。起動時刻は `ntp\grs\config\CronConfig.xml` ファイルの `CleanCacheReportExecutor` で設定でき

ます。夜間ジョブの構成の詳細については「レポート キャッシュのカスタマイズ」を参照してください。

- Report Center の管理者モードで、[ツール] メニューの [データ キャッシュ] > [キャッシュの無効化] を選択します。
- DTP Server を再起動します。

注意！

- 特定のレポートがキャッシュされ、Parasoft ツールからの新しいログを Data Collector が受け取るとき、このレポート キャッシュは無効化されません。レポートはキャッシュから表示されます。そのため、新しいログをレポートで参照するには、キャッシュをメニューから無効化するべきです。そうしないと、レポート キャッシュが午前零時過ぎに自動的に実行されます。
- レポートのキャッシュ データは dtp/reportcenter/datacache ディレクトリに格納されます。特定のパラメーターを持つレポートごとに、2 種類のファイルが格納されます。1 つは .dat ファイルであり、もう 1 つは .par ファイルです。

レポート キャッシュのカスタマイズ

レポート キャッシュは次の 2 箇所カスタマイズすることができます。

- dtp/grs/config/CronConfig.xml 構成ファイル。このファイルは Report Center のバックグラウンド ジョブの設定を定義します。
キャッシュに関連するのは次の 2 つのジョブです。
 - CacheReportExecutor:
dtp/reportcenter/staticLinksConfig.xml 構成ファイルで定義されたレポートをブラウザしてキャッシュするバックグラウンド ジョブ。このジョブはデフォルトでは毎朝 5 時に実行されます。
 - CleanCacheReportExecutor:
レポート キャッシュを無効化するジョブ。デフォルトでは、毎晩 12:01 に実行されます。
- Report Center の管理者ページで [ツール] メニューの [データ キャッシュ] > [Cache Settings] を選択します。
 1. キャッシュのオン / オフを切り替えます。
 2. Report サーバーのアイドル時間を分で指定します。レポート キャッシュを自動生成する CacheReportExecutor は、停止した後にこのアイドル時間が過ぎると、再び動作し始めます。

CacheReportExecutor は CronConfig.xml で定義された時刻 (デフォルトは allowFromHour=5 a.m.) に開始し、次のいずれかの状況が発生するまで動作します。

- staticLinksConfig.xml で指定されたすべてのレポートがキャッシュされた場合。
- CacheReportExecutor が動作中にユーザーが Report Center レポートを参照し始め、CacheReportExecutor が一時停止し、すべてのユーザーが Report Center レポートの参照を終了した後に、Report サーバーのアイドル時間が経過して CacheReportExecutor が再び動作した場合。

- CronConfig.xml で定義された allowToHour に CacheReportExecutor が達した場合 (staticLinksConfig.xml で定義されたすべてのレポートがキャッシュされたとき、それまでに停止していなければ、この時点で CacheReportExecutor は停止します)。

Report Center のツール

Report Center には、chk、send、および size ツールが用意されています。このセクションでは各ツールの使い方について説明します。

スクリプトの抽出とカスタマイズ

Report Center のツールを使用するには、SDLCEExtensions-`{version}`.tgz または SDLCEExtensions-`{version}`.exe をシステム中のディレクトリに展開します。

例：

UNIX: `$HOME/proserve`

Windows: `C:\Program Files\Parasoft\ProserveTools`

次のスクリプトが含まれます。

UNIX スクリプト：

- `chk.sh` は Report Center 中のデータをチェックします。
- `send.sh` は Report Center にテスト メッセージを送信します。
- `size.sh` は Report Center のサイズをチェックし、その情報を Report Center に格納します。

Windows スクリプト：

- `chk.cmd` は Report Center 中のデータをチェックします。
- `send.cmd` は Report Center にテスト メッセージを送信します。
- `size.cmd` は Report Center のサイズをチェックし、その情報を Report Center に格納します。

注意：通常、これらのスクリプトを適切に実行するには、スクリプトの内容を変更する必要があります。多くの場合、スクリプトがインストールされているシステムに特有のパス情報に合わせてスクリプトを変更します。スクリプトはローカル ディレクトリからだけ実行されるように作成されていますが、パスから実行するように簡単に変更できます。

UNIX スクリプトのカスタマイズ

シェル スクリプトを編集します。通常、`java` がインストールされている場所を指すよう、`JAVA_HOME` を変更する必要があります。

bash ユーザーの場合：`export JAVA_HOME=/usr/local/java`

tcsh ユーザーの場合：`setenv JAVA_HOME/usr/local/java`

`export JAVA_HOME=/usr/local/java`

自分のパスから実行する場合、"set GTDIR" の行をコメントアウトした後、GTDIR にインストールディレクトリのフルパスを設定する行を追加する必要があります。

```
#export GTDIR=`pwd`
export GTDIR=$HOME/proserve/ReportCentertools
```

Windows スクリプトのカスタマイズ

注意： C:\Program Files\Parasoft\ProserveTools にインストールしていて、Report Center も同じシステム上にインストールしている場合、スクリプトを変更する必要はありません。

cmd ファイルを編集します。通常、java がインストールされている場所を指すよう、JAVA_HOME を変更する必要があります。

```
set JAVA_HOME=C:\j2sdk5.0
```

自分のパスから実行する場合、"For /F "tokens=*" の行をコメントアウトした後、GTDIR にインストールディレクトリのフルパスを設定する行を追加する必要があります。

```
REM For /F "tokens=*" %%v in ('CD') Do @Set GTDIR=%%v
set GTDIR=C:\Program Files\Parasoft\ProserveTools
```

ツールの実行

chk.sh | chk.cmd (v2.1) の実行

データベース中のデータをチェックする chk ツールは、いくつかの方法で実行できます。デフォルトでは、現在の日付と 1 日前の日付のデータだけがチェックされます。基本的な使用では、次のようにコマンドに Report Center サーバーの名前を指定します。

UNIX:

```
./chk.sh reportcenter_server_name
```

Windows:

```
chk reportcenter_server_name
```

Source Scanner のデータをチェックするには、コマンドに `-scanner` パラメーターを追加します。

UNIX:

```
./chk.sh reportcenter_server_name -scanner
```

Windows:

```
chk reportcenter_server_name -scanner
```

Report Center のデータをローカル ディレクトリに CSV ファイルとしてダンプするには、次のコマンドを使用します。

UNIX:

```
./chk.sh reportcenter_server_name -dump -all
```

Windows:

```
chk reportcenter_server_name -dump -all
```

Oracle ユーザーのための注意事項 :

Report Center サーバーがデフォルトの MySQL サーバーではなく Oracle db を使用している場合、上記のコマンドに "-oracle" を追加します。

例 :

```
chk reportcenter_server_name -oracle
```

send.sh | send.cmd (v1.2) の実行

Data Collector にデータを送信する send.sh または send.cmd ツールを実行するには、データを送信する Report Center サーバーの名前をコマンドで指定します。

UNIX:

```
./send.sh reportcenter_server_name
```

Windows:

```
send reportcenter_server_name
```

size.sh | size.cmd (v2.1.1) の実行

Report Center データベースのサイズをチェックする size.sh | size.cmd ツールを実行するには、次のようにコマンドと Report Center サーバーの名前を入力します。

UNIX:

```
./size.sh reportcenter_server_name
```

oracle の場合 :

```
./size.sh -oracle reportcenter_server_name  
to do local size check use "-f"  
./size.sh -f reportcenter_server_name
```

Windows:

```
size reportcenter_server_name
```

oracle の場合 :

```
size -oracle reportcenter_server_name  
      to do local size check use "-f"  
size -f reportcenter_server_name
```

Team Server サンドボックスの作成

Team Server に格納されるテスト データやレポートが増加するにつれて、データ セットを明確に分けることが重要になります。

Team Server でサンドボックスを使用すると、異なる開発グループごとに論理的にデータを分離できます。また、バックアップと復元の処理を他と分ける方法を拡張できるだけでなく、権限を制御する方法も拡張できます。

チーム テスト コンフィギュレーションはテンプレートとしてデフォルト エリアに格納されます。チーム リーダーはここからカスタム チーム サンドボックスを作成できます。C++test などのツールの Team Server プロパティは、デフォルトではユーザー名もパスワードも設定されていません。そのため、チーム テスト コンフィギュレーションを変更したり Team Server のデフォルト エリアにテスト結果をアップロードしたりすることが可能です。

Team Server の管理

Team Server を管理するには、次のタスクを完了します。

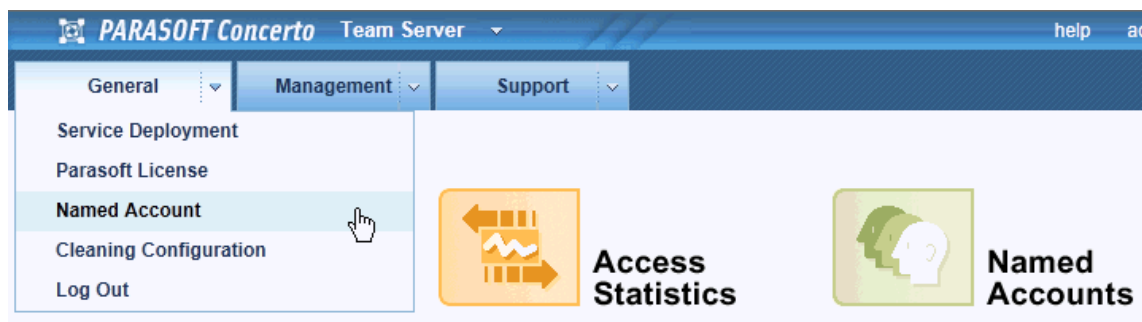
- Team Server のアドミン アカウントを作成する
- 新しいサンドボックスを作成する
- チーム テスト コンフィギュレーションをチームのサンドボックスにロードする

Team Server のアドミン アカウントを作成する

Team Server をロックダウンしてアドミン アカウントを作成します。

1. Team Server ホーム ページ に移動し、[全般] メニューの [名前付きアカウント] をクリックします。

Team Server ホーム ページ



2. [Enable unauthorized connections] チェックボックスをオフにして [Save] をクリックします。

[Named accounts configuration] ページ

PARASOFT Development Testing Platform Team Server

General Management Support

Named accounts configuration page

[Add new account]

Modify existing accounts

[Manage all accounts' permissions]

Enable unauthorized connections. [Manage permissions]

[Save]

Login	Active	Permissions	Remove account
BankExample	<input checked="" type="checkbox"/>	Manage	Remove
dtp	<input checked="" type="checkbox"/>	Manage	Remove
test	<input checked="" type="checkbox"/>	Manage	Remove
webtest_admin	<input checked="" type="checkbox"/>	Manage	Remove

3. [Add new account] をクリックしてグローバル アドミン アカウントを追加します。
4. 次のように各フィールドに入力して [Add account] をクリックします。
 - **Username:** admin
 - **Password:** password
 - **Path prefix:** 空白のままにします。
5. ルート ファイル システムへのアクセスを検証します。
 - a. [管理] メニューの [データ ストレージ] をクリックします。
 - b. [Manage path permissions] をクリックして、アドミンおよび未認証ユーザーの次の設定を確認します。

Permissions management ページ

[Back](#)

Permissions management page

Path: /

Accounts list:

- [admin](#)
- [unauthorized accounts](#)

Permissions for account: admin [\[Save\]](#) [\[Back To Top\]](#)

Path: /

Can modify:

Permissions for unauthorized accounts [\[Save\]](#) [\[Back To Top\]](#)

Path: /

Can modify:

Note: value *not set* means that permission is not specified for this path. Permissions for parent paths will be used.

新しいサンドボックスを作成する

新しいプロジェクトを作成する必要がある場合、チーム リーダーは Team Server の管理者に連絡を取って次の作業を行う必要があります。下記の説明では、チーム名を team1 とします。

1. Team Server ホーム ページ に移動し、[Named Accounts] アカウントをクリックするか、[全般] メニューの [名前付きアカウント] をクリックします。
2. [Add new account] をクリックして、チームのアドミン アカウントとユーザー アカウントを追加します。
 - チームのアドミン アカウントの設定
 - **Login:** team1_admin
 - **Password:** 任意のパスワード
 - **Path prefix:** team1
 - [Add account] をクリックします。

team1_admin アカウントの追加

[Back](#)

Named account configuration page

Login:

Password:

Retype password:

Path prefix: (max 15 chars)

[\[Add account\]](#)

- チームのユーザー アカウントの設定 :
 - **Login:** team1_user
 - **Password:** 任意のパスワード
 - **Path prefix:** team1
 - [\[Add account\]](#) をクリックします。

team1_user アカウントの追加

[Back](#)

Named account configuration page

Login:

Password:

Retype password:

Path prefix: (max 15 chars)

[\[Add account\]](#)

3. 各チーム アカウントの権限を設定します。

- チーム アドミン アカウント (この例では team1_admin) の設定
 - [Named accounts configuration] ページで、設定するアカウントに対応する [Permissions] 列の [Manage] リンクをクリックします。
 - [Path:/] の [can modify] のドロップダウン リストを [false] に変更します。
 - [Add New Path] をクリックしてデフォルト パスを入力します。たとえば /usr/team1 と入力します。
 - [can modify] を [true] にして [Add] をクリックします。
 - [Save] をクリックしてすべてのユーザー設定を保存します。

team1_admin の設定

[Back](#)

Permissions management page

Permissions for account: team1_admin [\[Save\]](#)

All permissions should be set for paths which starts with **/usr/team1** (except default one - "/")

Path: /
 Can modify:

[\[Add New Path\]](#)

Note: value *not set* means that permission is not specified for this path. Permissions for parent paths will be used.

Adding permissions for user: team1_admin.

All permissions should be set for paths which starts with **/usr/team1**

New path:

Can modify:

[\[Add\]](#) [\[Cancel\]](#)

- チーム ユーザー アカウント (この例では team1_user) を設定します。
 - [Manage] をクリックします。
 - [Path:/] の [Can modify] を [false] に変更します。
 - [Save] をクリックしてすべてのユーザー設定を保存します。

注意: クライアント ツールのデータ (テスト構成、レポート、抑制など) が保存されるパスはツールおよびバージョンによって異なります。データごとにアップロードの制御を行いたい場合は、各データや保存されるパスごとに設定を行ってください。データや保存されるパスについては、サポートまでご確認ください。

team1_user の設定

[Back](#)

Permissions management page

Permissions for account: team1_user [\[Save\]](#)

All permissions should be set for paths which starts with **/usr/team1** (except default one - "/")

Path: /
 Can modify:

[\[Add New Path\]](#)

Note: value *not set* means that permission is not specified for this path. Permissions for parent paths will be used.

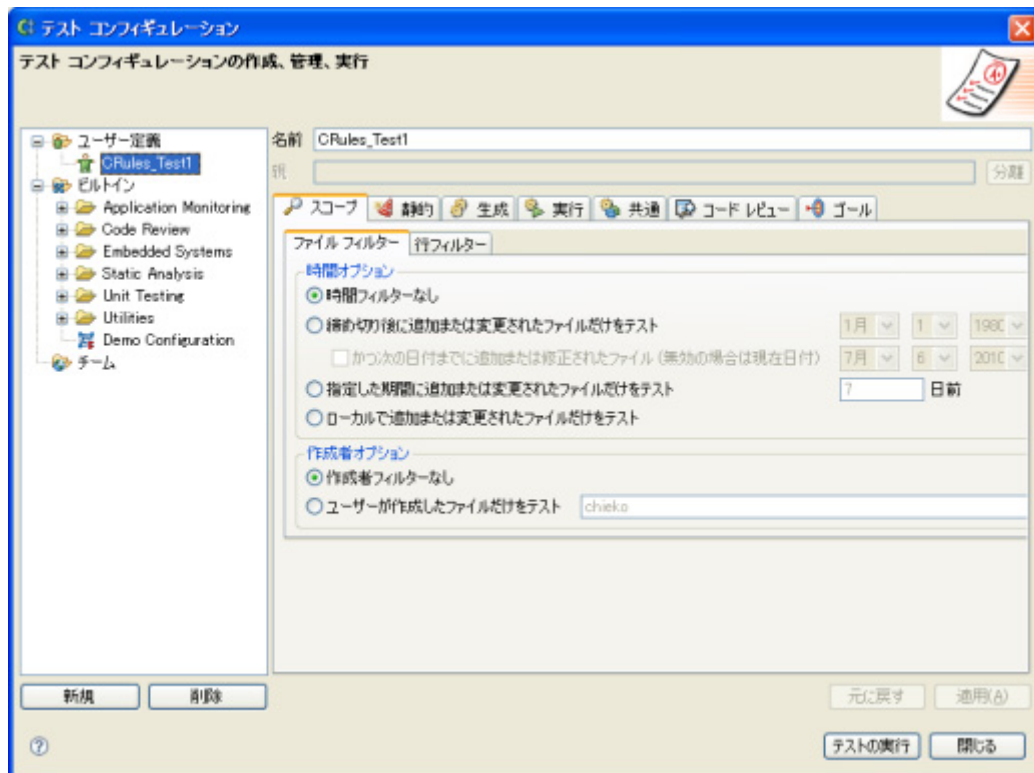
チーム テスト コンフィギュレーションをチームのサンドボックスにロードする

作成したサンドボックスにはテスト コンフィギュレーションが 1 つもロードされていないため、チーム リーダーまたは Team Server の管理者がチームのデフォルト コンフィギュレーションをロードする必要があります。次の操作を行います。

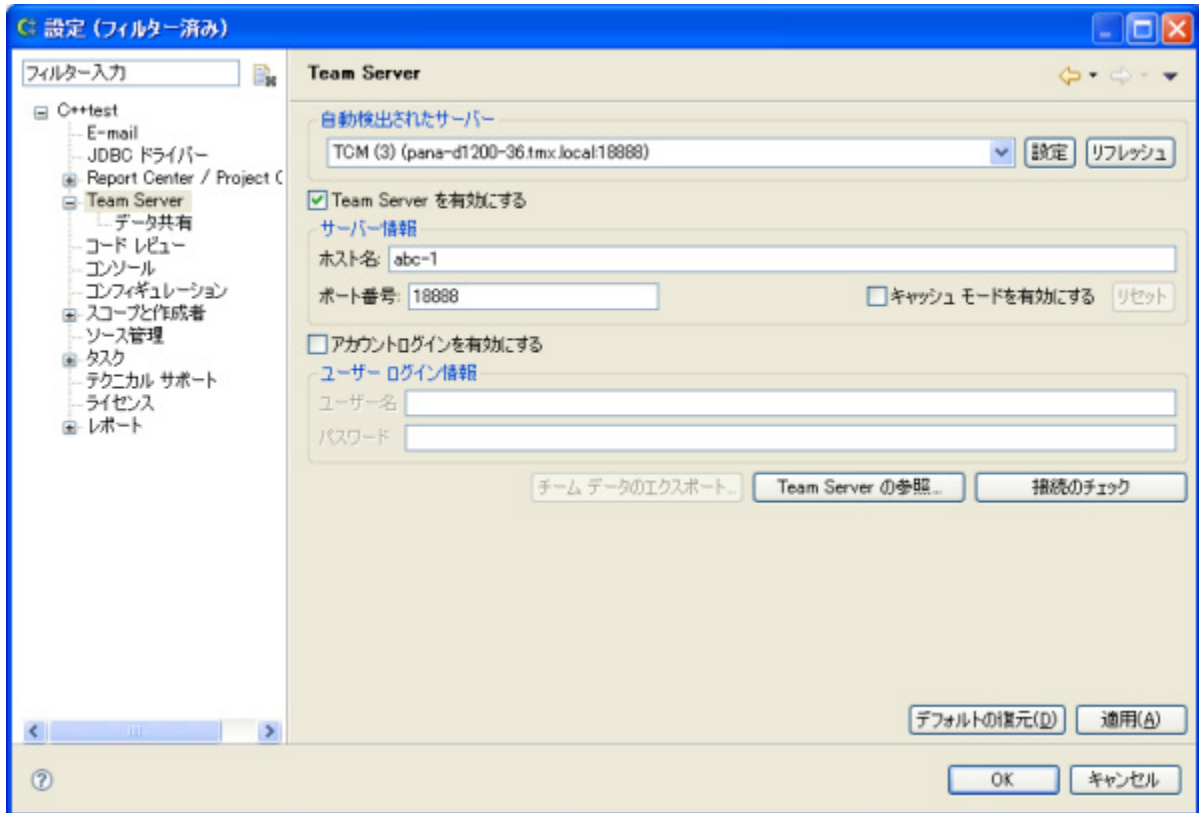
以下の操作説明では、例として C++test を使用しています。

1. C++test などの適切なツールを起動します。
2. [C++test] メニューの [テスト コンフィギュレーション] をクリックします。
3. [チーム] カテゴリのテスト コンフィギュレーションから適切なテスト コンフィギュレーションを選択し、右クリックしてショートカット メニューの [複製] をクリックします。

選択したチーム テスト コンフィギュレーションが [ユーザー定義] カテゴリにコピーされます。

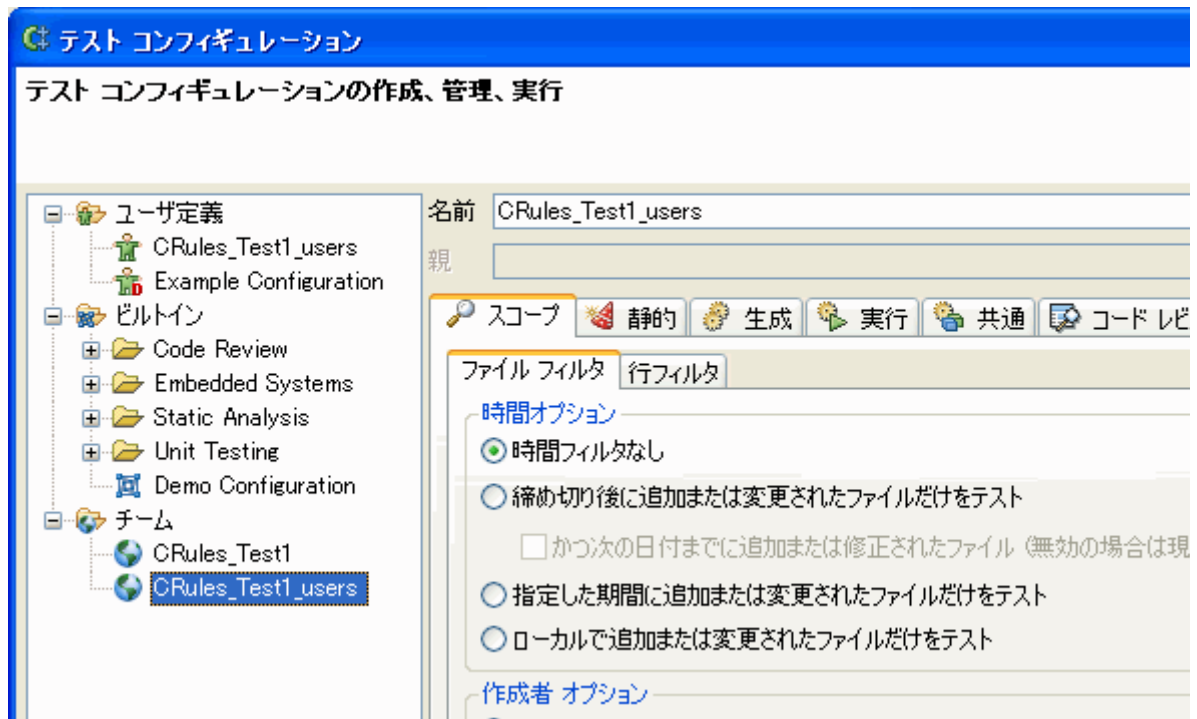


4. [名前] フィールドで、コピーしたテスト コンフィギュレーションの名前を変更します。[適用] ボタンをクリックしてから [閉じる] ボタンをクリックしてこのユーザー定義テスト コンフィギュレーションを保存します。
5. [C++test] メニューの [設定] をクリックします。
6. 左側のリストから [Team Server] を選択します。
7. [Team Server を有効にする] チェックボックスをオンにし、[ホスト名] フィールドにチームの Team Server のホスト (ホスト名または IP アドレス) を入力します。デフォルト (18888) 以外のポート番号を使用している場合、[ポート番号] フィールドにチームの Team Server のポート番号を入力します。
8. (オプション) ユーザー認証を行う場合、[アカウント ログインを有効にする] チェックボックスをオンにし、Team Server の管理者から指定された ユーザー名とパスワードを入力します。
9. [接続のチェック] ボタンをクリックして Team Server への接続を検証し、[OK] をクリックします。
10. [適用] ボタンをクリックしてから [OK] をクリックして設定を保存します。



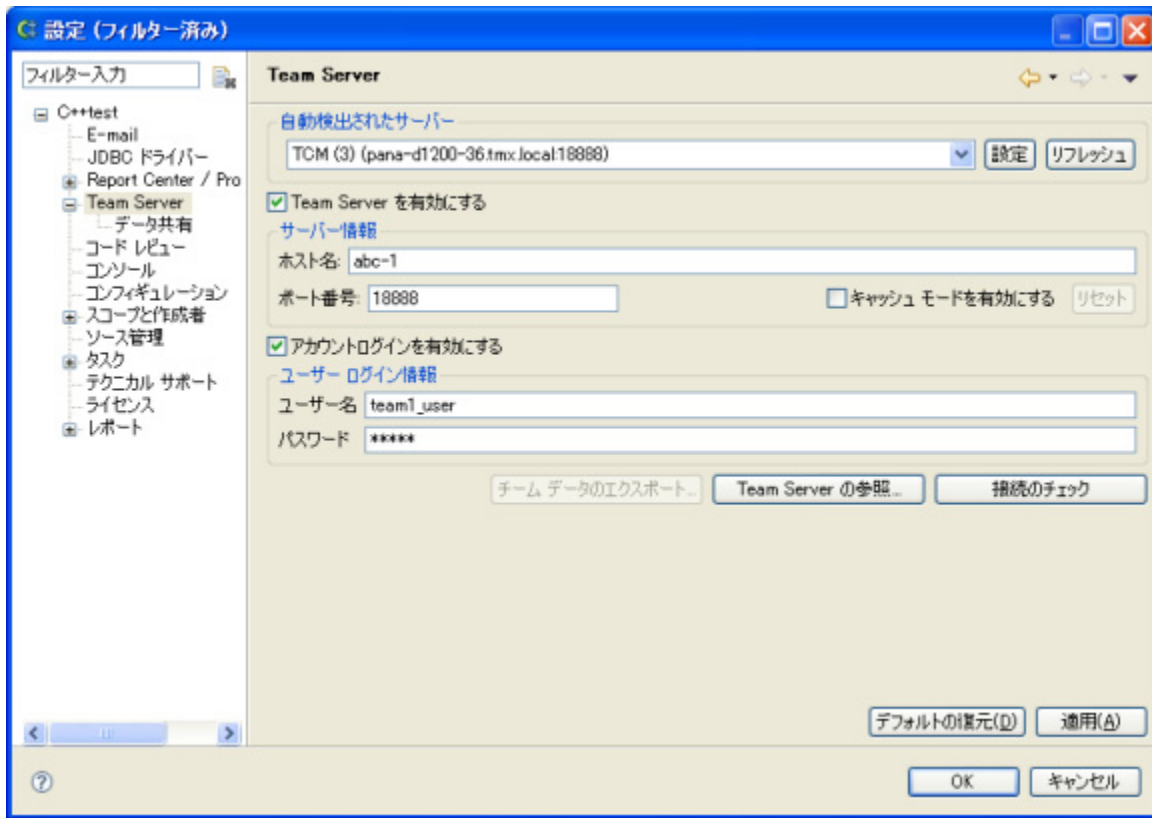
11. 再度 [C++test] メニューの [テスト コンフィギュレーション] をクリックします。
12. Team Server に追加するユーザー定義テスト コンフィギュレーションを右クリックし、ショートカット メニューの [Team Server へアップロード] をクリックします。

選択したユーザー定義テスト コンフィギュレーションのコピーが作成されて [チーム] カテゴリの下に表示されます。



重要な注意事項

- チーム リーダーまたは Team Server の管理者から指定された Team Server に接続するとき、チーム メンバーである開発者は、チーム ユーザー アカウントを使用する必要があります。



- C++test などのツールをコマンドライン実行またはサーバー実行する場合、ツールの実行データを正しいサンドボックスに確実にアップロードするために、ログイン設定にチームアドミンのログイン情報を使用する必要があります。
- Team Server の管理者は、Team Server に保管されたデータをバックアップするプロセスを設定する必要があります。
- Team Server の消去の設定は「30 日ごと」に設定する必要があります。この設定を行うには、Team Server ページで [全般] メニューの [Cleaning Configuration] をクリックします。消去する時刻は、[time] フィールドで適切な時刻を設定してください。また、設定を変更するたびに必ず [Save] をクリックしてください。

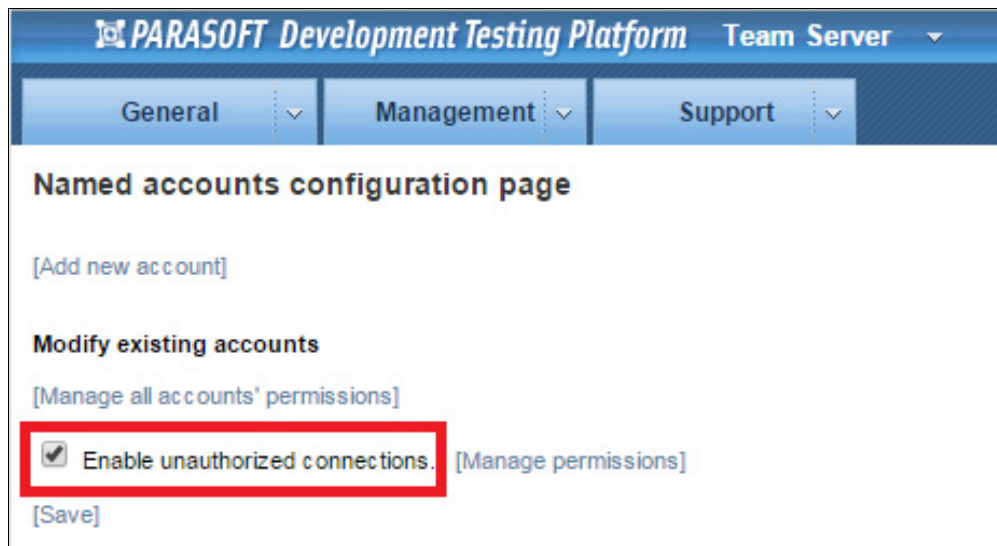
Data Collector から Team Server への DTP Engine 10.x レポートの転送

DTP Engine 10.x の静的解析レポートを Team Server に転送するように DTP を設定できます。そうすれば、DTP Engine が検出した静的解析違反を Parasoft Test 9.x のデスクトップ IDE にダウンロードできます。DTP Engine 10.x (DTP Engine for C/C++ など) のビルドサーバーと Parasoft Test 9.x (Parasoft C/C++test 9.6 など) のデスクトップマシンを組み合わせで使用している場合、この設定が役に立つでしょう。

1. DTP_HOME\conf\PSTRootConfig.xml 構成ファイルを開きます。
2. <root-config> タグの下に以下のエントリを追加します。

```
<enable-forward-to-team-server>true</enable-forward-to-team-server>
```

Team Server の **[Enable unauthorized connections]** オプションが有効な場合、Data Collector が受け取る DTP Engine の静的解析レポートはすべて Team Server にデフォルトで転送されます。



[Enable unauthorized connections] オプションが無効で、名前付きアカウントを使用している場合 (692 ページの「名前付きアカウント (Named Accounts) の定義」を参照)、プロジェクトの [Parasoft Test グローバル設定] または [Parasoft Test 設定] でユーザー名とパスワードを指定する必要があります。基本的な使用方法は次のとおりです。

```
tcm.server.username=team_user
```

```
tcm.server.password=123456789
```

詳細については『Parasoft Test ユーザーズ ガイド』を参照してください。

マルチキャスト DNS の使用

Development Testing Platform (DTP) はマルチキャスト DNS を使ってローカル ネットワーク上の Parasoft Test や他の DTP Server にサービスをブロードキャストします。そのため、Parasoft Test は DTP Server を自動的に検出することができ、その結果、Parasoft Test の設定が非常に容易になります。

マルチキャスト DNS をそのまま有効にしておくことを強く推奨します。ただし、必要に応じて無効にすることもできます。

マルチキャスト DNS を無効化するには、次の操作を行います。

1. <DTP インストール ディレクトリ >/conf/pcc.conf ファイルを開き、<Root> ノードの子として <jmdns-autoconf-disabled>>true</jmdns-autoconf-disabled> ノードを追加します。

ファイルは次のようになるはずです。

```
<Root>
  <license>
    ...
  </license>
  <jmdns-autoconf-disabled>>true</jmdns-autoconf-disabled>
  ...
...
```

2. <DTP インストール ディレクトリ >/tomcat/webapps/bpel/WEB-INF/engine.xml ファイルを開き、advertiseEndpoint オプションを false に設定します。

Microsoft Word へのエクスポートのカスタマイズ

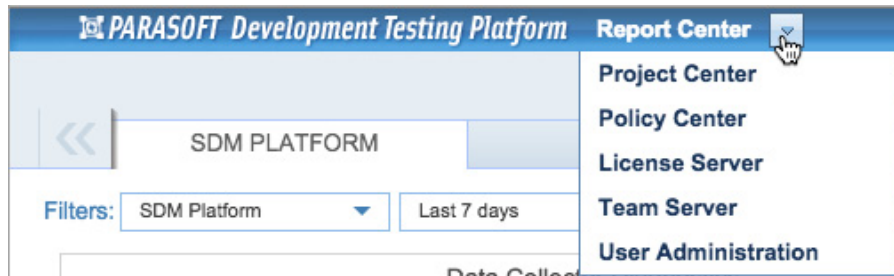
一部のレポートには、Microsoft Word 文書 (docx) にエクスポートするオプションがあります。[Export to Word] ボタンをクリックすると、現在のレポートが docx ファイルに開きます。この docx ファイルには、Development Testing Platform (DTP) のヘッダーとフッターが定義された Microsoft Word テンプレートが使用されます。

たとえば会社独自のテンプレートなど、カスタム テンプレートを使用する場合は次の操作を行います。

1. カスタム テンプレート ファイル (.docx) を `$DTP_HOME/grs/xreports/planning/common/docx` ディレクトリにコピーします。
2. ファイル名を `template.docx` に変更します。

ツールバーでのアプリケーションの無効化と有効化

Report Center や Project Center といったアプリケーションを DTP のツールバーで無効化することができます。



1. [INSTALLATION_DIR]\conf\PSTRootConfig.xml 構成ファイルをエディターで開きます。
2. <visible-apps> セクションのコメントを外し、ツールバーで無効化するアプリケーションには false を、有効化するアプリケーションには true を設定します。

<visible-apps> セクションが存在しない場合、<root-config> ノードの下に手動で追加する必要があります。

```
<!-- Configures which web application can be visible in DTP head menu -->
```

```
<visible-apps>
  <report-center>true</report-center>
  <project-center>true</project-center>
  <policy-center>true</policy-center>
  <team-server>true</team-server>
  <license-server>true</license-server>
  <user-administration>true</user-administration>
</visible-apps>
```

3. 編集が完了したら、DTP Server を再起動する必要があります。

各アプリケーションのライセンスも必要です。また、個々のユーザーは有効化されたアプリケーションを見るための権限が必要です (603 ページの「[User Administration] ページ」を参照してください)。

Development Testing Platform の統合

このセクションの内容：

- Parasoft Test から Report Center/Project Center へのテスト結果の送信
- CSV からの BTS データのインポート
- BTS/RMS Scanner と Updater
- Report Center および Project Center とサードパーティ製ツールの統合
- DTP イベントのサブスクリプション

Parasoft Test から Report Center/ Project Center へのテスト結果の送信

Jtest、C++test、dotTEST、SOAtest といった Parasoft 製品からテスト結果をインポートするには、接続パラメーターを適切に設定する必要があります。

1. Parasoft Test 製品で [Parasoft] メニューの [設定] をクリックします。
2. [Parasoft] > [Development Testing Platform] を選択します。
3. 適切な接続プロパティを指定します。詳細については『Parasoft Test ユーザーズ ガイド』の「Parasoft Concerto および Parasoft Concerto Report Center との接続」を参照してください。
4. [適用] をクリックします。
5. [OK] をクリックして設定を保存します。

CSV からの BTS データのインポート

DTP は、CSV ファイル (カンマ区切りファイル) を使用して任意のバグ トラッキング システムと統合することができます。バグ トラッキング システムが CSV ファイルにエクスポートできる欠陥 / 改善であれば、DTP にインポートできます。

CSV ファイルを使用して DTP と BTS/RMS を統合するには、以下のセクションで説明する手順を実行します。

- CSV ファイルの準備
- CSV ファイルの書式
- BTS Scanner 構成ファイルの準備

注意: CSV ファイルが、たとえば欠陥の名前などに national characters を使用している場合、DTP にインポートする前にファイルを UTF-8 エンコーディングで保存してください。

また、.csv、.xls、または .xlsx ファイルから DTP のプロジェクトに内部的な欠陥をインポートすることもできます。ファイルから欠陥をインポートすると、DTP はそれを内部的な欠陥として認識します。その結果、より多くのプロパティが定義できるようになります。詳細については 337 ページの「ファイルからの欠陥のインポート」を参照してください。

CSV ファイルの準備

CSV を使用して DTP と BTS/RMS を統合するには、次の 2 つのファイルが必要です。

- **CSV_bugs.csv:** BTS の欠陥のリストを含むファイルです。
- **CSV_activities.csv:** 欠陥のステータスの変更履歴を含むファイルです。

DTP\grs\extras\csv-bts にファイルのサンプルがあります。

BTS Scanner ジョブは定期的にこの 2 つのファイルを読み込んでデータを DTP に送信します (デフォルトの設定では 20 分ごとです。または DTP の Web インターフェイスで指定された値です)。

DTP の欠陥データが最新の状態に保たれるよう、ファイルを定期的に更新すること (つまり BTS からデータをエクスポートすること) を推奨します。一日一回、あるいは理想的には BTS で変更が行われるたびにファイルを更新することを推奨します。

以下のセクションでは BTS 統合で使用する CSV ファイルの作成方法について説明します。

CSV 欠陥ファイルの作成

DTP は CSV 欠陥ファイルから欠陥 / 改善の基本的な情報を読み取ります。例として、ファイル名が CSV_bugs.csv であると仮定します。

A	B	C	D	E	F	G	H	I
ID	Type	Summary	Status	Created On	Owner	Priority	Severity	Resolution
40174	defect	Problem with add-ins.	VERIFIED	12/21/09 06:26 AM	user2	medium	Low	FIXED
40185	enhancement	Reorganize categories in HTML Report	NEW	12/21/09 12:26 PM	user4	high		
40175	defect	Unit testing for Z project fails.	RESOLVED	12/21/09 07:06 AM		medium	High	FIXED
40176	defect	[internal] Update plug-ins	RESOLVED	12/21/09 09:49 AM	user5	medium	Medium	FIXED

DTP がこのファイルを定期的にスキャンする点に注意してください。

どのデータを CSV_bugs.csv ファイルにエクスポートするかについては、次の 2 種類の方法が考えられます。

- CSV_bugs.csv ファイルを定期的に削除し、最近変更された欠陥や新しく追加された欠陥の中で、DTP にインポートしたい欠陥だけを保存します。

たとえば、すでに DTP にインポート済みの欠陥のうち 1 つの欠陥のステータスが変更され、同時にまだ DTP にインポートされていない別の欠陥が 1 つ BTS に追加されている場合、CSV_bugs.csv ファイルには 2 行のデータがあり、この 2 つの欠陥に関する最新の情報が記載されます。

- BTS のすべての欠陥を CSV_bugs.csv ファイルに書き出します。DTP は定期スキャンを実行する際、最終スキャン以降に変更または追加された欠陥の情報だけを更新します。

注意： DTP にインポートされた欠陥は永久に削除されません。CSV_bugs.csv ファイルを更新することで、欠陥のプロパティの更新と新たな欠陥の追加だけを行うことができます。

CSV_bugs.csv ファイルの列：

以下のプロパティは必須です。

- **ID** バグ トラッキング システムでの欠陥 / 改善の識別子。
- **Type** 項目が欠陥か改善かを表します。この列の値は、「ix) <bts>/<feature-request>」で説明する <feature-request> タグに指定された項目のいずれかと一致する場合にだけフィーチャーと解釈されます。一致しない場合、値は欠陥と解釈されます。
- **Summary** 欠陥の概要。
- **Status** 欠陥のステータスが open か closed かを表します。「vi) <bts>/<resolved-status>」も参照してください。
- **Created On** BTS で欠陥が作成された日

以下のプロパティは DTP レポートに表示されるため、CSV ファイルに含めることを強く推奨します。

- **Owner** 現在 BTS で欠陥に割り当てられている担当者
ヒント： Owner プロパティには DTP のログイン名か、DTP の Users Administration モジュールに定義された E-mail アドレスと一致するユーザー名を使用することを推奨します。そうすることで、Project Center モジュールで欠陥 / 改善からタスクをスケジュールし、DTP の所有者に割り当てるのが容易になります。
- **Priority/Severity** 欠陥の優先度
- **Resolution** 欠陥が解決された方法。未解決の場合は空にしておきます。「vii) <bts>/<inactive-resolution>」も参照してください。

以下のプロパティ (列) を CSV_bugs.csv ファイルに含めることができます。

Version、Milestone、Hardware、Os、Project、および Component

これらのプロパティは将来の使用に備えて DTP にインポートされます。次の場所にこれらのプロパティを含むサンプル ファイルがあります。

DTP\grs\extras\csv-bts\CSV_bugs_ext.csv

CSV 欠陥履歴ファイルの作成

DTP は CSV 欠陥履歴ファイルから欠陥 / 改善に加えられたステータス変更の履歴を読み取ります。例として、ファイル名が CSV_activities.csv であると仮定します。

注意： ステータス以外のフィールドの変更をファイルに含めることもできますが、それらの情報は現在のバージョンの DTP では使用されません。

CSV_bugs.csv ファイルの場合と同じように、どのデータを CSV_activities.csv ファイルにエクスポートするかについて 2 種類の方法が考えられます。

- CSV_activities.csv ファイルを定期的に削除し、ファイルには DTP にインポートしたい最近行われたステータス変更だけを保存します。
- すべてのステータス変更履歴を CSV_activities.csv ファイルに書き出します。DTP は定期スキャンを実行する際、最終スキャン以降に追加された情報だけを更新します。

ステータス変更履歴は Report Center レポートに表示されます。たとえば、Tester Dashboard はプロジェクトの欠陥が時間の経過につれてどのように変化したかという傾向を表すグラフを表示します。

このファイルに `_gresolution_h` プロパティを保存することもできます。このプロパティは特定の欠陥が非アクティブ状態にされたかどうかを指定するのに使われます。

ID	User	Activity Date	Changed Field	Old Value	New Value
40174	user5	01/17/10 06:05 AM	Status	NEW	RESOLVED
40174	user5	01/25/10 01:18 AM	Status	RESOLVED	VERIFIED
40175	user5	01/23/10 07:03 AM	Status	NEW	ASSIGNED
40175	user1	01/23/10 07:44 AM	Status	ASSIGNED	RESOLVED
40176	user5	01/26/10 09:50 AM	Status	NEW	RESOLVED

CSV_activities.csv ファイルの列：

履歴ファイルには以下の情報を含める必要があります。

- **ID** 欠陥の ID
- **User** 変更を行ったユーザー
- **Activity Date** 変更が行われた日
- **Changed Field** 変更された属性。実際に DTP レポートで使用されるのはステータス属性ですが、所有者や解決方法など他の属性を含めて DTP にインポートすることもできます。
- **Old Value** 変更前のフィールドの値
- **New Value** 変更後のフィールドの値

CSV ファイルの書式

CSV ファイルには以下の書式ルールを適用します。

- 各値をカンマで区切ります。値にカンマが含まれる場合、引用符で囲みます。
- 引用符はダブルクォーテーションを使用します。
- 行の最後のフィールドの後にはカンマを付けません。

- Null または空フィールドは 2 つのカンマで表します。
- 通常、ファイルの末尾は 1 つの END-OF-LINE 文字で終わります。

BTS Scanner 構成ファイルの準備

DTP BTS Scanner ジョブの構成ファイルが必要です。

構成ファイルのサンプルが次の場所にあります。

DTP\grs\config\bts\examples\ExampleCSVScannerConfig.xml

ファイルをコピーし、以下で説明するすべてのパラメーターを変更してください。

変更が完了したら、ファイルを DTP\grs\config\bts ディレクトリに置きます。DTP はこのフォルダーを定期的にスキャンし、発見した構成ファイルに基づいて BTS をスキャンします。

BTS Scanner で CSV ファイルをスキャンするには、以下の属性値を指定します。

i) `<bts>/<name>`

BTS を識別するためにユーザーが見て分かる名前を指定します。

例 :My BTS Server

この名前は Project Center の [欠陥 / 改善検索] ページに [欠陥トラッキング システム] ドロップダウン リストの選択肢として表示されます。

ii) `<bts>/<bugs-file>`

CSV_bugs.csv ファイルの場所

iii) `<bts>/<activities-file>`

CSV_activities.csv ファイルの場所

iv) `<bts>/<fields-mapping>`

このセクションでは CSV_bugs.csv ファイルの列名を指定します。ID から Created On 列は必須です。Owner から Severity 列を指定することを強く推奨します。残りの列はオプションであり、将来の使用に備えて DTP にインポートされます。

```
<fields-mapping>
  <id>ID</id>
  <bug-type>Type</bug-type>
  <summary>Summary</summary>
  <status>Status</status>
  <creation-date>Created On</creation-date>
  <assigned-to>Owner</assigned-to>
  <priority>Priority</priority>
  <severity>Severity</severity>
  <resolution>Resolution</resolution>

  <version>Version</version>
  <milestone>Milestone</milestone>
  <hardware>Hardware</hardware>
```

```

    <os>Os</os>
    <project>Project</project>
    <component>Component</component>
</fields-mapping>

```

v) **<bts>/<activity-fields-mapping>**

このセクションでは CSV_activities.csv ファイルの列名を指定します。すべての列が CSV_activities.csv 中に存在する必要があります。

```

<activity-fields-mapping>
  <id>ID</id>
  <who>User</who>
  <date>Activity Date</date>
  <changed-field>Changed Field</changed-field>
  <old-value>Old Value</old-value>
  <new-value>New Value</new-value>
</activity-fields-mapping>

```

vi) **<bts>/<resolved-status>**

欠陥が解決されたことを表す値を指定します。つまり、欠陥の修正または改善の実装を表す値を指定します。

この情報は DTP のさまざまなレポートで使用されます。たとえば、Tester Dashboard グラフは、オープン中の欠陥と未テスト / テスト済みの欠陥の割合が時間の経過につれどのように変化したかを表示します。クローズ済みの欠陥が未テスト / テスト済みの欠陥として扱われます。

例：

```

<resolved-status>
  <status>RESOLVED</status>
  <status>CLOSED</status>
</resolved-status>

```

vii) **<bts>/<inactive-resolution>**

欠陥が非アクティブであることを表す値を指定します。非アクティブの欠陥は DTP にインポートされません。

例：

```

<inactive-resolution>
  <resolution>DUPLICATE</resolution>
  <resolution>INVALID</resolution>
</inactive-resolution>

```

viii) **<bts>/<date-formats>**

CSV_bugs.csv および CSV_activities.csv ファイルで使われる日付書式を指定します。最も一般的に使用される書式があらかじめ指定されています。

```

<date-formats>
  <date-format>yyyy-MM-dd hh:mm</date-format>
  <date-format>yyyy-MM-dd hh:mm:ss.0</date-format>
  <date-format>yy MMM-dd hh:mm</date-format>
  <date-format>MM/dd/yy hh:mm aa</date-format>
</date-formats>

```

書式は Java の SimpleDateFormat クラスに定義された書式のいずれかでなければなりません (<http://java.sun.com/javase/6/docs/api/java/text/SimpleDateFormat.html> を参照)。

ix) <bts>/<feature-request>

<item> ノードの値は、CSV_bugs.csv ファイルの _gType_h 列の値のうち、欠陥ではなく改善として扱う値を指定します。

例：

```
<feature-request>
  <item>enhancement</item>
</feature-request>
```

インポートされた CSV 欠陥 / 改善の検証方法

CSVScannerConfig.xml の変更が完了したら、ファイルを DTP\grs\config\bts ディレクトリに置きます。DTP はこのフォルダーを定期的にスキャンし、見つかった構成ファイルに基づいて BTS をスキャンします。DTP が BTS 構成ファイルをスキャンする方法の詳細については 275 ページの「BTS Scanner の実行」を参照してください。BTS Scanner または RMS Scanner を構成する方法は 2 つあります。

バグ トラッキング システムおよび要求管理システムの統合

Development Testing Platform (DTP) は、BTS Scanner または RMS Scanner を使用して、バグ トラッキング システム (BTS) や要求管理システム (RMS) と統合します。BTS Scanner と RMS Scanner は内部的なコンポーネントであり、DTP と統合する BTS/RMS ごとに構成する必要があります。BTS Scanner と RMS Scanner は、サポートされるシステムからデータベースに、バグと要求のデータをそれぞれ転送します。

各 Scanner は、BTS または RMS のデータベースから重要な情報を取得して、Report Center データベースに送ります。転送されたデータは、バグまたは要求に関連するレポートを作成するために Report Center と Project Center が使用します。構成が完了すると、BTS/RMS Scanner はデフォルトの 15 分間隔で自動的に実行されます。また、必要なときに手動で実行することもできます。

ビルトイン UI を使った BTS/RMS Scanner の構成

Report Center または Project Center の管理ページから BTS Scanner と RMS Scanner を構成することができます。この構成は以下のシステムについてだけ行うことができます。

バグ トラッキング システム

- Bugzilla
- JIRA 4.x
- JIRA 5.x/6.x (DTP Server で実行するには SSL が必要。詳細については 758 ページの「SSL を使用した Development Testing Platform の実行」を参照)

要求管理システム

- Blueprint
- DOORS

BTS 構成の追加

1. 管理ページを開き、[設定] > [BTS Scanners] を選択します。
2. [タイプ] ドロップダウン メニューからシステムを選択します。

3. 新しい構成の名前を入力し、[新規作成] をクリックします。

The screenshot shows the 'BTS Scanner Configurations' page. At the top, there are navigation tabs for 'Main', 'Projects', and 'Settings'. Below the tabs, the page title is 'BTS Scanner Configurations'. Underneath, it says 'Create a new configuration'. There is a 'Type' dropdown menu set to '--- Bug Tracking Systems ---' and a 'Name' text input field. A 'Create New' button is located below the input fields. Below the form, there is a table with 4 items. The table has columns for 'Modify', 'Migrate', 'Name', 'BTS ID', 'Type', and an unlabeled column. The rows contain configuration details for 'asdf', 'BZ test scanner config', 'Jira 5', and 'test'.

Modify	Migrate	Name	BTS ID	Type	
Modify	Migrate to JIRA 5.x	asdf	31	JIRA 4.x	JIRA-13-07-10_0
Modify		BZ test scanner config	32	Bugzilla	BUGZILLA-13-10
Modify		Jira 5	33	JIRA 5.x	JIRA5-13-04-22_
Modify	Migrate to JIRA 5.x	test	34	JIRA 4.x	JIRA-13-07-10_0

BTS Scanner の構成













1. BTS Scanner の管理ページを開き、[修正] 列の [修正] をクリックします。
2. サーバー設定とログイン情報を入力します。

The screenshot shows the 'Edit BTS Scanner Configuration' page. At the top, there are navigation tabs for 'Main', 'Projects', and 'Settings'. Below the tabs, the page title is 'Edit BTS Scanner Configuration'. Underneath, it says 'JIRA 5.x configuration'. There are two main sections: 'Scanner Setting' and 'Development Testing Platform Setting'. The 'Scanner Setting' section has fields for 'Name' (Jira 5), 'BTS ID' (32), 'JIRA URL' (http://jira2.parasoft.com), 'Username' (devtest), and 'Password' (masked with dots). A 'Check' button is below these fields. The 'Development Testing Platform Setting' section has fields for 'DTP Username' (admin) and 'DTP Password' (masked with dots). A 'Check' button is below these fields.

3. 成果物のステータスを DTP にマッピングする方法を設定します。

JIRA statuses that are mapped to Concerto

6 Item(s)


Status		
	Closed	
	Open	
	In Progress	
	Reopened	
	Duplicate	
	Won't Fix	

4. 設定が完了したら [保存] をクリックします。

RMS 構成の追加

1. 管理ページを開き、[設定] > [RMS Scanners] を選択します。
2. [タイプ] ドロップダウン メニューからシステムを選択します。
3. 新しい構成の名前を入力し、[新規作成] をクリックします。

Main Projects Settings

RMS Scanner Configurations 

Create a new configuration

Type:

Name:


2 Item(s)

Modify	Name	RMS ID	Type	File Name
Modify	Blueprint 2	5	Blueprint	BLUEPRINT-13-05-23_043747.x
Modify	My Blueprint	6	Blueprint	BLUEPRINT-13-06-07_035049.x

RMS Scanner の構成

1. RMS Scanner の管理ページを開き、[修正] 列の [修正] をクリックします。

2. サーバー設定とログイン情報を入力します。

 **Edit RMS Scanner Configuration**

Blueprint configuration

Scanner Setting

Name: ?

RMS ID:

Blueprint URL: ?

Username:

Password:

Development Testing Platform Setting

DTP Username: ?

DTP Password:

Import Settings

Target Project: ?

UI による IBM Rational DOORS の構成

DOORS の構成ページには、そのシステムに固有のパラメーターが含まれます。

1. 上記の新しい構成を追加する操作手順に従います。
2. RMS Scanner の構成ページを開き、[修正] 列の [修正] をクリックします。
3. サーバー設定とログイン情報を入力します。

4. [追加] ボタンをクリックし、要求を DTP Server にインポートする DOORS モジュールを定義します。指定するパスは、DOORS のフォーマル モジュールまたはリンク モジュールを指す必要があります。たとえば DOORS フォルダーを指してはいけません。

DOORS configuration

General

Name: DOORS rms scanner test

BTS ID: Database Error

DOORS exe: c:\Program Files\IBM\Rational\DOORS\9.2\bin\doors.x

User: Administrator

Password: qaz123

Modules to scan

3 Item(s) Add

Delete	Module name*	DTP project*	DOORS DB Example: 29998@localhost	User	Password
<input checked="" type="checkbox"/>	/Test/Requirements/System Requirements	Default Project			
<input checked="" type="checkbox"/>	/Sports utility vehicle 4x2/Requirements/Functional Requirements	Default Project	29998@localhost	Eric McCall	Training
<input checked="" type="checkbox"/>	/Sports utility vehicle 4x2/Requirements/User Requirements	Default Project	29998@localhost	Eric McCall	Training

5. さらに DOORS の統合情報を指定します。
- **DTP プロジェクト**：指定した DOORS モジュールの要求をインポートする DTP プロジェクト。
 - **DOORS DB**：(任意) 代替 DOORS データベース。
 - **注意**：対応するモジュール名が、DOORS Exe にリンクされている以外の DOORS db で定義されている場合にだけ (General 構成で指定)、このフィールドを使用します。
 - **ユーザー**：DOORS データベースにアクセスでき、Module Name のすべてのオブジェクトに対する読み取り権限を持った有効な DOORS ユーザー名。
 - **パスワード**：User の有効なパスワード。
6. [保存] をクリックします。

新しい DOORS BTS Scanner 構成ファイルが \$DTP_HOME/grs/config/bts ディレクトリに作成されます。

手動による BTS/RMS Scanner の構成

DTP_HOME/grs/config/bts ディレクトリに格納された XML ファイルを編集して、BTS Scanner と RMS Scanner を手動で構成することができます。BTS Scanner と RMS Scanner の構成と使用方法については 272 ページの「BTS/RMS Scanner と Updater」を参照してください。

DTP はいくつかのシステムとの単純な統合をサポートします。以下のセクションでは、BTS/RMS Scanner を手動で構成し、各システムと連携して使用方法について説明します。

- HP Quality Center との統合
- Bugzilla との統合

- IBM Rational ClearQuest との統合
- Atlassian JIRA との統合
- IBM Rational Change および Rational Synergy との統合

BTS/RMS Scanner と Updater

手動で BTS Scanner を構成し、BTS Scanner の .xml 構成ファイルを編集して BTS/RMS を BTS Scanner でスキャンすることができます。この .xml 構成ファイルは、DTP Server がインストールされているホストの `DTP_HOME/grs/config/bts` に格納されています。

この章では次のトピックについて説明します。

- サンプル構成ファイルの準備
- 構成ファイルの追加
- カスタム BTS Scanner/Updater
- BTS Scanner の実行
- BTS Scanner の動作確認
- BTS Updater の構成
- オリジナル ID の使用

サンプル構成ファイルの準備

`DTP_HOME/grs/config/bts/examples` フォルダーには、サポート対象 BTS/RMS ごとにサンプル構成ファイルが用意されています。

サンプル構成ファイルを準備するには、次の操作を行います。

1. サンプル構成ファイルのディレクトリに移動します。
`DTP_HOME/grs/config/bts/examples`
2. 使用している BTS 用のサンプル ファイルを選択します。
たとえば、BTS Scanner を JIRA と統合するには、次のファイルを選択します。
`ExampleJiraScannerConfig.xml`
3. 選択したサンプル ファイルを同じディレクトリにコピーします。
`DTP_HOME/grs/config/bts/examples`
4. ファイル名を変更します。
たとえば、`ExampleJiraScannerConfig.xml` を `MyJiraScannerConfig.xml` に変更します。
5. 構成ファイルを編集し、ベンダーに合わせて BTS/RMS 設定を指定します。

BTS Scanner は現時点でいくつかの BTS/RMS をサポートしていますが、それぞれ必要な構成は異なります。

BTS Scanner と特定の BTS/RMS を連携するための構成の詳細については、以下を参照してください。

- HP Quality Center との統合
- Bugzilla との統合
- IBM Rational ClearQuest との統合
- Atlassian JIRA との統合 --older versions

BTS Scanner は CSV ファイルから欠陥をインポートすることもできるので、データを CSV ファイルにエクスポートする機能のある BTS/RMS であれば、DTP と統合できます。BTS Scanner で CSV ファイルをインポートする方法の詳細については、260 ページの「CSV からの BTS データのインポート」を参照してください。

ヒント: DTP では統合する成果物の名前をカスタマイズできます。BTS Scanner と BTS/RMS の統合をセットアップした後に、DTP で使う成果物名を BTS/RMS で使われている名前に合わせると便利です。詳細については 215 ページの「成果物の名前変更」を参照してください。

構成ファイルの追加

BTS/RMS に合わせて構成ファイルを修正したら、BTS Scanner で構成ファイルを読み込みます。BTS Scanner は定期的に行われ、`DTP_HOME/grs/config/bts` にある構成ファイル进行处理します。

新規に DTP をインストールした場合、`bts` ディレクトリは空です。このディレクトリには複数の構成ファイルを追加できます。そのため、BTS Scanner は複数の異なる種類の BTS/RMS に接続することも、同じ種類でパラメーターの異なる複数の BTS/RMS インスタンスに接続することもできます。`bts` ディレクトリに追加された各構成ファイルは、それぞれ 1 つの BTS/RMS インスタンスとの統合に相当します。

準備した構成ファイルを追加して BTS Scanner で読み込むには、次の操作を行います。

- 選択したファイルを `DTP_HOME/grs/config/bts` にコピーまたは移動します。

カスタム BTS Scanner/Updater

次の Java API を実装してカスタム BTS Scanner/Updater を作成することができます。

- `com.parasoft.api.dtp.defects.scan.DefectScanner`
- `com.parasoft.api.dtp.defects.scan.DefectScannerFactory`
- `com.parasoft.api.dtp.defects.update.DefectUpdater`
- `com.parasoft.api.dtp.defects.update.DefectUpdaterFactory`

Java API ドキュメントへのアクセスについては 337 ページの「Java API」を参照してください。ログに関する情報については 677 ページの「ログ ファイルの参照」を参照してください。

カスタム実装のインストール

1. カスタム BTS Scanner/Updater のための jar ファイルを作成します。
2. この jar ファイルを `$DTP_HOME/tomcat/webapps/grs/WEB-INF/lib` ディレクトリにコピーします。
3. DTP を再起動します。

カスタム実装を使用するよう BTS Scanner を構成する

\$DTP_HOME/grs/config/bts に XML 構成ファイルを作成する必要があります。テンプレートとして \$DTP_HOME/grs/config/bts/examples/ExampleCustomScannerConfig.xml を使用できます。

カスタム XML 構成ファイルで最も重要なのは次の 2 箇所です。

- scannerFactoryClass 要素の値は、
com.parasoft.api.dtp.defects.scan.DefectScannerFactory インターフェイスを実装するクラス名に設定しなければなりません。
- updatorFactoryClass 要素の値は、
com.parasoft.api.dtp.defects.update.DefectUpdaterFactory インターフェイスを実装するクラス名に設定しなければなりません。

BTS Scanner の統合についての既存ドキュメントはこの章にあります。

カスタム RMS Scanner

次の Java API を実装して、カスタムの RMS Scanner を作成することができます。

- com.parasoft.api.dtp.requirements.scan.RequirementScanner
- com.parasoft.api.dtp.requirements.scan.RequirementScannerFactory

Java API ドキュメントへのアクセスについては 337 ページの「Java API」を参照してください。ログに関する情報については 677 ページの「ログ ファイルの参照」を参照してください。

カスタム実装のインストール

1. カスタム RMS Scanner/Updater のための jar ファイルを作成します。
2. この jar ファイルを \$DTP_HOME/tomcat/webapps/grs/WEB-INF/lib ディレクトリにコピーします。
3. DTP を再起動します。

カスタム実装を使用するよう RMS Scanner を構成する

\$DTP_HOME/grs/config/bts に XML 構成ファイルを作成する必要があります。テンプレートとして \$DTP_HOME/grs/config/bts/examples/ExampleCustomRmsScannerConfig.xml を使用できます。

scannerFactoryClass 要素の値は、
com.parasoft.api.dtp.requirements.scan.RequirementScannerFactory インターフェイスを実装するクラス名に設定しなければなりません。これはカスタム XML 構成ファイルで最も重要な部分です。

BTS Scanner の実行

「オリジナル ID の使用」の説明に従って BTS/RMS との統合を構成し、.xml ファイルを \$DTP_HOME/grs/config/bts ディレクトリに保存すると、BTS Scanner がファイルを処理し、定義された BTS/RMS に接続して欠陥 / 要求に関する基本的な情報を DTP データベースにインポートします。

BTS Scanner は自動的に実行されますが、必要なときに手動で実行して BTS/RMS で行われた最新の変更を DTP のデータに反映することもできます。どちらの場合でも、BTS Scanner が期待どおりに動作していることを検証できます。

BTS Scanner の自動実行

BTS Scanner は DTP のバックグラウンド ジョブです。BTS Scanner は 15 分ごとに実行されます。\$DTP_HOME/grs/config/bts ディレクトリにある各構成ファイルで指定された BTS/RMS に接続し、データを DTP にインポートします。

BTS Scanner の実行間隔を設定することができます。次の操作を行います。

1. \$DTP_HOME/grs/config/CronConfig.xml ファイルの frequency パラメーターを変更します。

```
<Job frequency="15" id="BTS Scanner" runDayOfWeek="*"
runHour="*" run-Minute="*">

    <class
name="com.parasoft.grs.rserver.cronjobs.bts.BtsScannerJob"
"/>

</Job>
```

2. DTP Server を再起動します。

BTS Scanner の手動実行

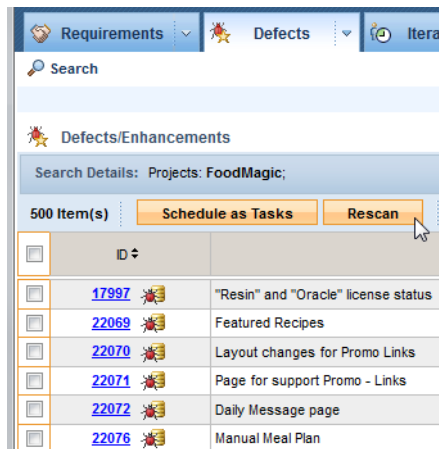
必要なときに DTP の Web インターフェイスから BTS Scanner を実行できます。以下のどちらかの場所から実行できます。

- Project Center: 欠陥 / 改善 の検索
詳細については 275 ページの「[欠陥 / 改善 の検索] ページから BTS Scanner を手動で実行する」を参照してください。
- Report Center: メイン > 管理者 > ツール > 欠陥 / 改善の計算
詳細については 200 ページの「スキャンと計算の自動実行」を参照してください。

[欠陥 / 改善 の検索] ページから BTS Scanner を手動で実行する

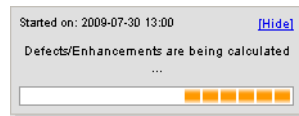
1. Project Center で [欠陥] > [検索] をクリックし、検索を実行します。
2. [欠陥 / 改善] ページの [再スキャン] ボタンをクリックします。

欠陥 / 改善の再スキャン



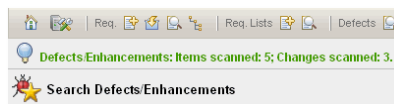
ダイアログが開き、スキャンの進捗が表示されます。

スキャン進行中メッセージ



スキャンが完了すると、ステータス バーにスキャンされたバグおよび変更の数が表示されます。

スキャン完了メッセージ



BTS Scanner の動作確認

BTS Scanner は、自動的に実行することも、必要なときに手動で実行することもできます。ユーザーは、BTS Scanner が実行中かどうか、またデータが期待どおりに DTP にインポートされたかを検証できます。

BTS Scanner のバックグラウンド作業を検証するには、DTP_HOME/logs ディレクトリにある `bts_scanner.log` ファイルを確認します (677 ページの「ログ ファイルの参照」を参照)。

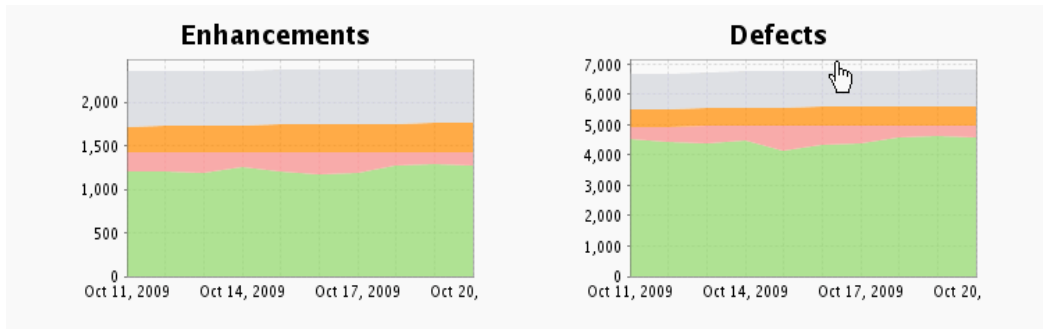
BTS Scanner が BTS/RMS からデータを DTP データベースにインポートしたかどうかを検証するには、以下のどちらかの操作を行います。

1. 欠陥 / 改善の検索

- [欠陥トラッキング システム] ドロップダウン メニューを展開します。
このメニューには DTP と統合されている BTS/RMS の名前が表示されます。名前は .xml 構成ファイルの `<bts/name>` ノードで定義した名前です。
- [検索] ボタンをクリックします。
DTP データベースにインポートされたすべての欠陥が検索されて表示されます。

欠陥 / 改善の検索に関する詳細については『Project Center ユーザーズ ガイド』の「欠陥 / 改善の検索」を参照してください。

2. Report Center で [テスト] > [変更ベースのテスト] > [要求 / 欠陥] を選択します。
3. デフォルト プロジェクトに切り替えます。DTP データベースのすべての欠陥 / 改善データが表示されます。



4. Report Center と Project Center で欠陥 / 改善に対応する方法については、以下のセクションを参照してください。
 - 185 ページの「ソース管理フィルター」
 - 144 ページの「欠陥および改善レポート」
 - 410 ページの「欠陥 / 改善」
 - 423 ページの「要求と欠陥 / 改善のタスク化」

BTS Updater の構成

DTP がサポートする BTS/RMS の場合、BTS/RMS 中の課題に関連付けられたタスクが変更されると、DTP は BTS/RMS 中のその課題のステータスとコメントを変更します。この処理は BTS Updater というユーティリティを使って行われます。BTS Updater は統合 / サポートされたバグトラッキングシステムに DTP の更新を伝播します。

注意： 現在 BTS Updater はすべてのバージョンの Bugzilla でコメントの更新をサポートします。ただし、ステータスの更新ができるのは Bugzilla 3.4 以上だけです。

- サポートされる Bugzilla のバージョンについては 294 ページの「Bugzilla との統合」を参照してください。
- DTP がバグトラッキングシステムとどのようにやり取りするのかについては 544 ページの「バグトラッキングシステムとの同期」を参照してください。

BTS Updater を設定して、BTS/RMS にステータスとコメントの変更を伝播することができます。それには、BTS Scanner の .xml 構成ファイルを適切に設定します。この構成ファイルは、DTP Server がインストールされているホスト上の `DTP_HOME/grs/config/bts` ディレクトリにあります。

BTS Updater を有効化するには、新しいユーザー名とパスワードを既存の Bugzilla .xml 構成ファイルの `<connection-settings>` ノードに追加します。

例：


```

<bts>
  <connection-settings>
    <user>root</user>
    <pass encrypted="false">root</pass>
  </connection-settings>
</bts>

```

重要！ このユーザーは、統合される BTS の Web インターフェイスに対するアカウントを持っていないければなりません。また、そのアカウントは BTS 中のすべてのプロジェクトに対して読み取り / 書き込み権限を持つべきです。

オリジナル ID の使用

[Requirements Code Review] および [Requirements Code Review Details] レポートで DTP の ID ではなくタスク / 要求のオリジナル ID を使用する場合、次の操作を行います。

1. `$DTP_HOME/grs/xreports/planning/req/RequirementsCodeReview.xml` ファイルを開きます。
2. `<private-parameters>` タグの下に次のエントリを追加します。
`<parameter name="useOriginalId">true</parameter>`
3. `DTP_HOME/grs/xreports/codereview/CodeReviewDetails.xml` ファイルにも同じエントリを追加します。

HP Quality Center

BTS Scanner は HP Quality Center (旧 Test Director) と連携するように構成できます。Test Director と Quality Center は同じ物を指しているため、このセクションではどちらも「QC」と呼びます。Quality Center の統合は、Quality Center 9、10、および 11 でテスト済みです。

BTS Scanner for QC を使用すると、QC データベースから Report Center に要求と欠陥をインポートできます。

DTP では 2 種類の方法で HP QC と統合することができます。

- HP QC データベースの直接アクセス。DTP は直接 QC データを読み取ってインポートします。
 - QC の要求は DTP に対する欠陥 (タイプは拡張) としてインポートされます。
 - QC の欠陥は DTP での欠陥としてインポートされます。
- オープン テスト アーキテクチャ (OTA) の使用。DTP は直接 QC データを読み取ってインポートします。
 - QC の要求は DTP に対する機能要求としてインポートされます。
 - QC の欠陥は DTP での欠陥としてインポートされます。

HP QC データベースへの直接アクセス

注意：ほとんどのユーザーは、ステップ 2 を除く下記のすべてのステップで、デフォルト設定を利用できます。QC データベースの接続設定は、ユーザー独自の接続に合わせてカスタマイズする必要があります。

HP QC データベースへの直接アクセスを介して統合を構成するには、次の操作を行います。

1. 以下の全般的な設定を行います。

設定	説明
<bts>/<type>	"TestDirector" に設定するべきです。
<bts>/<name>	QC サーバーを識別するためにユーザーが見て分かる名前を指定します。 例： "Test Director" QC サーバーの各インスタンスに固有の名前でなければなりません。この名前は Project Center の [欠陥 / 改善検索] ページに [欠陥トラッキング システム] ドロップダウン リストの選択肢として表示されます。

例：

```
<bts type="TestDirector">
<name>Test Director</name>
```

2. QC データベース接続設定を指定します。

データは QC プロジェクトごとに別個のデータベースに保存されます。そのため、スキャン対象として少なくとも 1 つのデータベース名を指定する必要があります。

各プロジェクトのデータベース名は Quality Center Site Administration または MS SQL クライアントから確認できます。

注意：定義されている QC データベース名を記録しておくことを推奨します。データベース接続を設定するときに QC データベース名が必要です。

QC との統合では以下の QC データベース接続設定が必要です。

設定	説明
<bts>/<db-connection>/<db-type>	MSSQL のみ
<bts>/<db-connection>/<user>	データベース ユーザー名
<bts>/<db-connection>/<pass>	データベース パスワード
<bts>/<db-connection>/<url>	QC データベース接続 URL 注意： connection url にデータベース名を含めないでください。
<bts>/<db-connection>/<database>	QC データベース名。少なくとも 1 つのデータベース名を指定します。

例：

```
<db-connection>
  <db-type>MSSQL</db-type>
  <!-- currently only MS SQL server supported. -->
  <user>sa</user>
  <!-- valid MS SQL server user -->
  <password encrypted="true">abc123</password>
  <!-- valid MS SQL server user passwd -->
  <url>jdbc:sqlserver://localhost;</url>
<!-- sample MSSQL connection url: <url>jdbc:sqlserver://HOST;</url> -->
```

注意： 接続 URL にはデータベース名を含めないでください。

```
<!-- At least one db name is required -->
<database>default_test_db</database>
<database>QualityCenter_Demo_db</database>
</db-connection>
```

3. フィールドマッピングを構成します。

欠陥と要求のフィールドマッピングは、データベースの BUG テーブルと REQ テーブルのカラム名に基づきます。各フィールドの詳細については、Quality Center のマニュアル (quality_center_db.chm) を参照してください。

欠陥および要求の field-mapping セクションはコメントアウトされています。セクションにはデフォルトのマッピング構成が記述されています。このセクションのコメントを削除して設定を変更できます。ただし、多くの場合、設定の変更は必要ありません。

注意： N/A はフィールドが利用できないことを表します。

例：

```
<defects>
  <fields-mapping>
    <summary>BG_SUMMARY</summary>
    <status>BG_STATUS</status>
    <resolution>BG_STATUS</resolution>
    <priority>BG_PRIORITY</priority>
    <severity>BG_SEVERITY</severity>
    <project>BG_PROJECT</project>
    <reporter>BG_DETECTED_BY</reporter>
    <assigned-to>BG_RESPONSIBLE</assigned-to>
    <creation-date>BG_DETECTION_DATE</creation-date>
    <version>BG_DETECTION_VERSION</version>
    <milestone>BG_PLANNED_CLOSING_VER</milestone>
    <hardware>N/A</hardware>
    <os>N/A</os>
    <component>BG_PROJECT</component>
    <modification-date>BG_VTS</modification-date>
  </fields-mapping>
</defects>

<requirements>
  <fields-mapping>
    <summary>RQ_REQ_NAME</summary>
    <status>RQ_REQ_STATUS</status>
```

```

        <resolution>RQ_REQ_STATUS</resolution>
        <priority>RQ_REQ_PRIORITY</priority>
        <severity>N/A</severity>
        <project>RQ_REQ_PRODUCT</project>
        <reporter>RQ_REQ_AUTHOR</reporter>
        <assigned-to>RQ_REQ_AUTHOR</assigned-to>
        <creation-date>RQ_REQ_DATE</creation-date>
        <version>N/A</version>
        <milestone>N/A</milestone>
        <hardware>N/A</hardware>
        <os>N/A</os>
        <component>RQ_REQ_TYPE</component>
        <modification-date>RQ_VTS</modification-date>
    </fields-mapping>
</requirements>

```

4. 「解決時のステータス」と「アクティブでない解決」の値を指定します。

「解決時のステータス」と「アクティブでないステータス」のマッピング構成は、`fields-mapping.status` の表記法に依存する場合があります。ただし、通常は設定の変更は必要ありません。

例：

```

<resolved-status>
    <!-- Requirement -->
    <status>Passed</status>
    <status>Failed</status>
    <status>Reviewed</status>
    <status>Not Completed</status>

    <!-- Defects -->
    <status>Closed</status>
    <status>Fixed</status>
</resolved-status>

<inactive-resolution>
    <!-- Requirement -->
    <resolution>N/A</resolution>

    <!-- Defects -->
    <resolution>Rejected</resolution>
</inactive-resolution>

```

注意：

- Report Center にインポートされた後、スキャンされた QC 項目に次の書式で識別子が付けられます。
 - PR の場合：BUG_(実際の QC ID)
 - FR の場合：REQ_(実際の QC ID)
- 現時点では、Report Center のレポートから QC の PR/FR 詳細を開くリンク機能は使用できません。
- Task Activity レポートにデータを表示するには、QC ユーザーを Report Center データベースに追加する必要があります。

- これはインクリメンタル スキャンです。つまり、前回のスキャンの後に QC で変更された要求 / 欠陥だけが DTP に対して更新されます。どの項目が QC で変更されたかを DTP が確実に把握できるようにするには、QC の Project Customization/Project Entities フォームで適切な設定を行う必要があります。ステップ 3 の「フィールド マッピングの構成」に記載されたすべてのフィールドは、アクティブな History プロパティを QC で持つべきです。

Windows 認証を使った Quality Center データベースへの接続

Windows 認証を介して MSSQL サーバーに接続するには、Windows 環境に DTP サービスがインストールされていて実行中でなければなりません。なぜなら、公式の MSSQL Server JDBC ドライバが使用する証明書は、Windows 認証のためのサービスを現在実行しているユーザー アカウントからの証明書だけだからです。

DTP のセットアップのために準備するには、次の操作を行います。

1. 公式の MSSQL Server JDBC ドライバ パッケージをダウンロードし、操作指示に従ってパッケージをアンロードします。
2. Windows のコントロール パネルを開き、[システムとセキュリティ] > [管理ツール] をクリックします。
3. [サービス] アイコンをダブルクリックします。
4. Parasoft Development Testing Platform を右クリックしてショートカット メニューの [プロパティ] をクリックします。
5. [ログオン] タブをクリックします。[アカウント] を選択し、LOCATION\username の書式で Windows ユーザー アカウントを入力します。このユーザー アカウントは MSSQL サーバー データベースにアクセスできなければなりません。
6. パスワードを入力し、[OK] をクリックします。
7. Parasoft Development Testing Platform を選択した状態で [サービスの停止] リンクをクリックし、DTP のサービスを停止します。
8. 新しいウィンドウで `${DTP_HOME}/tomcat/lib/thirdparty` ディレクトリを開きます。
9. `sqljdbc4.jar` ファイルのバックアップ コピーを作成します。
10. `<MSSQL_JDBC_DRIVER>\sqljdbc_<VERSION>\<LANG>` の `sqljdbc4.jar` を現行ディレクトリにコピーします。
11. `${DTP_HOME}/jre/bin` ディレクトリにアクセスし、`<MSSQL_JDBC_DRIVER>\sqljdbc_<VERSION>\<LANG>\auth\<ARCHITECTURE>` ディレクトリの `sqljdbc_auth.dll` を追加します。

DTP をセットアップするには、次の操作を行います。

1. `${DTP_HOME}/grs/config/bts` ディレクトリにアクセスし、QC BTS Scanner 構成ファイルを開きます。
2. QC BTS Scanner 構成ファイル中で `<db-connection>` 要素を探します。この構成ファイルがまだ設定されていない場合、286 ページの「HP QC データベースへの直接アクセス」の説明に従って適切に構成ファイルを変更します。

3. <db-connection> 要素内の <url> 要素の URL 値に対して、プロパティ `integratedSecurity=true` を追加します。
4. 以下の例のように設定して変更を保存します。

```
<db-connection>

<db-type>MSSQL</db-type>
<user>sa</user>
<password encrypted="true">abc123</password>
<url>jdbc:sqlserver://localhost;integratedSecurity=true;</url>

</db-connection>
```

5. [サービスの開始] リンクをクリックしてサービスの一覧から DTP サービスを開始します。
6. 276 ページの「BTS Scanner の動作確認」の説明に従って、QC Scanner が適切に構成されたことを確認します。

OTA (Open Test Architecture) の使用

OTA API を使用する BTS Scanner for QC を利用すると、QC データベースから DTP に要求と欠陥をインポートすることができます。QC の要求は Project Center の要求として、そして QC の欠陥は Project Center のバグとして表示されます。インポートされた要求と欠陥は読み取り専用モードであり、そのほとんどの属性を変更できません。

DTP がインストールされているのと同じマシンに QC クライアントがインストールされている場合、DTP から取得したデータを使って QC の要求を更新することもできます。

OTA (Open Test Architecture) を使って統合を構成するには、次の操作を行います。

1. 以下の一般的な設定を行います。

設定	説明
<bts>/<type>	"TestDirectorOTA" に設定するべきです。
<bts>/<name>	QC サーバーを識別するためにユーザーが見て分かる名前を指定します。 例: "Test Director" QC サーバーの各インスタンスに固有の名前でなければなりません。この名前は Project Center の [欠陥 / 改善検索] ページに [欠陥トラッキングシステム] ドロップダウン リストの選択肢として表示されます。

例:

```
<bts type="TestDirectorOTA">
<name>QC OTA</name>
```

Quality Center データベース接続設定を指定します。

データは QC プロジェクトごとに別個のデータベースに保存されます。このため、スキャン

対象として少なくとも 1 つのデータベース名を指定する必要があります (下記のサンプルの <database> タグを参照)。

各プロジェクトのデータベース名は Quality Center Site Administration または MS SQL クライアントから確認できます。

注意： 定義されている QC データベース名を書きとめておくことを推奨します。データベース接続を設定するときにこの情報が必要になります。

QC との統合では以下の QC データベース接続設定が必要です。

設定	説明
<bts>/<db-connection>/<db-type>	データベース ベンダー：MS SQL のみ。
<bts>/<db-connection>/<user>	データベース ユーザー。
<bts>/<db-connection>/<pass>	データベース パスワード。
<bts>/<db-connection>/<url>	Quality Center データベース接続 URL。 注意： データベース接続 URL にはデータベース名を含めないでください。
<bts>/<db-connection>/<database>	QC データベース名。少なくとも 1 つのデータベース名が必要です。 さらに <database> タグの次の属性を指定する必要があります。 <ul style="list-style-type: none"> • qc-domain – qc-project の QC ドメインを指定します。 • qc-project – QC プロジェクトの名前を指定します。 • dtp-project – DTP プロジェクトの名前を指定します。スキャンされた要求はこのプロジェクトの下に格納されます。 注意： スキャン プロセス中、指定されたプロジェクトが DTP データベース中にない場合、プロジェクトは自動的に作成されます。要求を適切に作成するために、qc-domain、qc-project、および dtp-project 属性が使用されます。

例：

```
<db-connection>
<db-type>MSSQL</db-type>
<user>sa</user>
<password encrypted="true">abc123</password>
<url>jdbc:sqlserver://localhost;</url>
<database qc-domain="DEFAULT" qc-project="QualityCenter_Demo"
dtp-project="MyProject">QualityCenter_Demo_db
</database>
</db-connection>
```

2. ota-connection を設定します。

DTP のデータで QC の欠陥を自動的に更新できるようにするには、OTA API を使ったやり取りを可能にする証明書を指定する必要があります。それには、以下の全般的な設定を行います。

設定	説明
<bts>/<ota-connection>/<user>	OTA API を使ってやり取りできるユーザーの名前。
<bts>/<ota-connection>/<password>	指定されたユーザーのための有効なパスワード。
<bts>/<ota-connection>/<url>	QC への有効な HTTP URL。

例：

```
<ota-connection>
<user>sa</user>
<password>sa</password>
<url>http://localhost:8080/qcbin</url>
</ota-connection>
```


HP Quality Center との統合

BTS Scanner を設定して HP Quality Center (以前の Test Director) と連携させることができます。Test Director と Quality Center は同義であるため、このセクションでは両方を指して "QC" と呼びます。Quality Center の統合は、Quality Center 9、10、および 11 でテスト済みです。

BTS Scanner for QC を使用すると、QC データベースから Report Center に要求と欠陥をインポートすることができます。

Development Testing Platform (DTP) では 2 種類の方法で HP QC と統合することができます。

- HP QC データベースの直接アクセス。DTP は直接 QC データを読み取ってインポートします。
 - QC の要求は DTP に対する機能要求としてインポートされます。
 - QC の欠陥は DTP での欠陥としてインポートされます。
- オープン テスト アーキテクチャ (OTA) の使用。DTP は直接 QC データを読み取ってインポートします。
 - QC の要求は DTP に対する機能要求としてインポートされます。
 - QC の欠陥は DTP での欠陥としてインポートされます。

HP QC データベースへの直接アクセス

注意：ほとんどのユーザーは、ステップ 2 を除く下記のすべてのステップで、デフォルト設定を利用できます。QC データベースの接続設定は、ユーザー独自の接続に合わせてカスタマイズする必要があります。

HP QC データベースへの直接アクセスによって統合を設定するには、次の操作を行います。

1. 以下の全般的な設定を行います。

設定	説明
<bts>/<type>	"TestDirector" に設定する必要があります。
<bts>/<name>	HP QC サーバーを識別するためにユーザーが見て分かる名前を指定します。 例: "Test Director" HP QC サーバーの各インスタンスに固有の名前でなければなりません。この名前は Project Center の [欠陥 / 改善検索] ページに [欠陥トラッキング システム] ドロップダウン リストの選択肢として表示されます。

例：

```
<bts type="TestDirector">
<name>Test Director</name>
```

2. QC データベース接続設定を指定します。

データは QC プロジェクトごとに別個のデータベースに保存されます。そのため、スキャン対象として少なくとも 1 つのデータベース名を指定する必要があります。

各プロジェクトのデータベース名は Quality Center Site Administration または MS SQL クライアントから確認できます。

注意：定義されている QC データベース名を記録しておくことを推奨します。データベース接続を設定するときに QC データベース名が必要になります。

QC との統合では以下の QC データベース接続設定が必要です。

設定	説明
<bts>/<db-connection>/<db-type>	MSSQL のみ
<bts>/<db-connection>/<user>	データベース ユーザー名
<bts>/<db-connection>/<pass>	データベース パスワード
<bts>/<db-connection>/<url>	QC データベース接続 URL 注意： connection url にデータベース名を含めないでください。
<bts>/<db-connection>/<database>	QC データベース名。少なくとも 1 つのデータベース名を指定します。

例：

```
<db-connection>
  <db-type>MSSQL</db-type>
  <!-- currently only MS SQL server supported. -->
  <user>sa</user>
  <!-- valid MS SQL server user -->
  <password encrypted="true">abc123</password>
  <!-- valid MS SQL server user passwd -->
  <url>jdbc:sqlserver://localhost;</url>
<!-- sample MSSQL connection url: <url>jdbc:sqlserver://HOST;</url> -->
```

注意：接続 URL にデータベース名を含めてはいけません。

```
<!-- At least one db name is required -->
<database>default_test_db</database>
<database>QualityCenter_Demo_db</database>
</db-connection>
```

3. フィールドマッピングを構成します。

欠陥と要求のフィールドマッピングは、データベースの BUG テーブルと REQ テーブルのカラム名に基づきます。各フィールドの詳細については、Quality Center のマニュアル (quality_center_db.chm) を参照してください。

欠陥および要求の field-mapping セクションはコメントアウトされています。セクションにはデフォルトのマッピング構成が記述されています。このセクションのコメントを削除して設定を変更できます。ただし、多くの場合、設定の変更は必要ありません。

注意： N/A はフィールドが利用できないことを表します。

例：

```
<defects>
  <fields-mapping>
    <summary>BG_SUMMARY</summary>
    <status>BG_STATUS</status>
    <resolution>BG_STATUS</resolution>
    <priority>BG_PRIORITY</priority>
    <severity>BG_SEVERITY</severity>
    <project>BG_PROJECT</project>
    <reporter>BG_DETECTED_BY</reporter>
    <assigned-to>BG_RESPONSIBLE</assigned-to>
    <creation-date>BG_DETECTION_DATE</creation-date>
    <version>BG_DETECTION_VERSION</version>
    <milestone>BG_PLANNED_CLOSING_VER</milestone>
    <hardware>N/A</hardware>
    <os>N/A</os>
    <component>BG_PROJECT</component>
    <modification-date>BG_VTS</modification-date>
  </fields-mapping>
</defects>

<requirements>
  <fields-mapping>
    <summary>RQ_REQ_NAME</summary>
    <status>RQ_REQ_STATUS</status>
    <resolution>RQ_REQ_STATUS</resolution>
    <priority>RQ_REQ_PRIORITY</priority>
    <severity>N/A</severity>
    <project>RQ_REQ_PRODUCT</project>
    <reporter>RQ_REQ_AUTHOR</reporter>
    <assigned-to>RQ_REQ_AUTHOR</assigned-to>
    <creation-date>RQ_REQ_DATE</creation-date>
    <version>N/A</version>
    <milestone>N/A</milestone>
    <hardware>N/A</hardware>
    <os>N/A</os>
    <component>RQ_REQ_TYPE</component>
    <modification-date>RQ_VTS</modification-date>
  </fields-mapping>
</requirements>
```

4. 「解決時のステータス」と「アクティブでない解決」の値を指定します。

「解決時のステータス」と「アクティブでないステータス」のマッピング構成は、fields-mapping.status の表記法に依存する場合があります。ただし、通常は設定の変更は必要ありません。

例：

```
<resolved-status>
  <!-- Requirement -->
  <status>Passed</status>
  <status>Failed</status>
```

```

<status>Reviewed</status>
<status>Not Completed</status>

<!-- Defects -->
<status>Closed</status>
<status>Fixed</status>
</resolved-status>

<inactive-resolution>
<!-- Requirement -->
<resolution>N/A</resolution>

<!-- Defects -->
<resolution>Rejected</resolution>
</inactive-resolution>

```

注意：

- Report Center にインポートされた後、スキャンされた QC 項目に次の書式で識別子が付けられます。
 - PR の場合：BUG_(実際の QC ID)
 - FR の場合：REQ_(実際の QC ID)
- 現時点では、Report Center のレポートから QC の PR/FR 詳細を開くリンク機能は使用できません。
- Task Activity レポートにデータを表示するには、QC ユーザーを Report Center データベースに追加する必要があります。
- これはインクリメンタル スキャンです。つまり、前回のスキャンの後に QC で変更された要求 / 欠陥だけが DTP に対して更新されます。どの項目が QC で変更されたかを DTP が確実に把握できるようにするには、QC の Project Customization/Project Entities フォームで適切な設定を行う必要があります。ステップ 3 の「フィールド マッピングの構成」に記載されたすべてのフィールドは、アクティブな History プロパティを QC で持つべきです。

Windows 認証を使った Quality Center データベースへの接続

Windows 認証を介して MSSQL サーバーに接続するには、Windows 環境に DTP サービスがインストールされていて実行中でなければなりません。なぜなら、公式の MSSQL Server JDBC ドライバが使用する証明書は、Windows 認証のためのサービスを現在実行しているユーザー アカウントからの証明書だけだからです。

DTP のセットアップのために準備するには、次の操作を行います。

1. 公式の MSSQL Server JDBC ドライバ パッケージをダウンロードし、操作指示に従ってパッケージをアンロードします。
2. [スタート] メニューを開き、[コンピューター] を右クリックして [管理] をクリックします (別の方法としては、コントロール パネルを開き、[システムとセキュリティ] > [管理ツール] をクリックします)。
3. [サービスとアプリケーション] を展開し、[サービス] を選択して Windows サービスの一覧を表示します。
4. Parasoft Development Testing Platform を右クリックして [プロパティ] を選択します。

5. [ログオン] タブをクリックします。[アカウント] を選択し、LOCATION\username の書式で Windows ユーザー アカウントを入力します。このユーザー アカウントは MSSQL サーバー データベースにアクセスできなければなりません。
6. パスワードを入力し、[OK] をクリックします。
7. Parasoft Development Testing Platform サービスを選択した状態で [サービスの停止] リンクをクリックし、DTP のサービスを停止します。
8. 新しいウィンドウで \${DTP_HOME}/tomcat/lib ディレクトリを開きます。
9. sqljdbc4.jar ファイルのバックアップ コピーを作成します。
10. <MSSQL_JDBC_DRIVER>\sqljdbc_<VERSION>\<LANG> の sqljdbc4.jar を現行ディレクトリにコピーします。
11. \${DTP_HOME}/jre/bin ディレクトリにアクセスし、<MSSQL_JDBC_DRIVER>\sqljdbc_<VERSION>\<LANG>\auth\<ARCHITECTURE> ディレクトリの sqljdbc_auth.dll を追加します。

DTP をセットアップするには、次の操作を行います。

1. \${DTP_HOME}/grs/config/bts ディレクトリにアクセスし、QC BTS Scanner 構成ファイルを開きます。
2. QC BTS Scanner 構成ファイル中で <db-connection> 要素を探します。この構成ファイルがまだ設定されていない場合、286 ページの「HP QC データベースへの直接アクセス」の説明に従って適切に構成ファイルを変更します。
3. <db-connection> 要素内の <url> 要素の URL 値に対して、プロパティ `integratedSecurity=true` を追加します。
4. 以下の例のように設定して変更を保存します。

```
<db-connection>

<db-type>MSSQL</db-type>
<user>sa</user>
<password encrypted="true">abc123</password>
<url>jdbc:sqlserver://localhost;integratedSecurity=true;</url>

</db-connection>
```

5. [サービスの開始] リンクをクリックしてサービスの一覧から Parasoft Development Testing Platform サービスを開始します。
6. 276 ページの「BTS Scanner の動作確認」の説明に従って、QC Scanner が適切に構成されたことを確認します。

OTA (Open Test Architecture) の使用

OTA API を使用する BTS Scanner for QC を利用すると、QC データベースから DTP に要求と欠陥をインポートすることができます。QC の要求は Project Center の要求として、そして QC の欠陥は Project Center のバグとして表示されます。インポートされた要求と欠陥は読み取り専用モードであり、そのほとんどの属性を変更できません。

DTP がインストールされているのと同じマシンに QC クライアントがインストールされている場合、DTP から取得したデータを使って QC の要求を更新することもできます。

OTA (Open Test Architecture) を使って統合を構成するには、次の操作を行います。

1. 以下の一般的な設定を行います。

設定	説明
<bts>/<type>	"TestDirectorOTA" に設定するべきです。
<bts>/<name>	QC サーバーを識別するためにユーザーが見て分かる名前を指定します。 例: "Test Director" QC サーバーの各インスタンスに固有の名前でなければなりません。この名前は Project Center の [欠陥 / 改善検索] ページに [欠陥トラッキングシステム] ドロップダウン リストの選択肢として表示されます。

例:

```
<bts type="TestDirectorOTA">
<name>QC OTA</name>
```

2. Quality Center データベース接続設定を指定します。

データは QC プロジェクトごとに別個のデータベースに保存されます。このため、スキャン対象として少なくとも 1 つのデータベース名を指定する必要があります。

各プロジェクトのデータベース名は Quality Center Site Administration または MS SQL クライアントから確認できます。

注意: 定義されている QC データベース名を書きとめておくことを推奨します。データベース接続を設定するときにこの情報が必要になります。

QC との統合では以下の QC データベース接続設定が必要です。

設定	説明
<bts>/<db-connection>/<db-type>	データベース ベンダー: MS SQL のみ。
<bts>/<db-connection>/<user>	データベース ユーザー。
<bts>/<db-connection>/<pass>	データベース パスワード。

設定	説明
<bts>/<db-connection>/<url>	Quality Center データベース接続 URL。 注意： データベース接続 URL にはデータベース名を含めないでください。
<bts>/<db-connection>/<database>	QC データベース名。少なくとも 1 つのデータベース名が必要です。 さらに <database> タグの次の属性を指定する必要があります。 <ul style="list-style-type: none"> • qc-domain – qc-project の QC ドメインを指定します。 • qc-project – QC プロジェクトの名前を指定します。 • dtp-project – DTP プロジェクトの名前を指定します。スキャンされた要求はこのプロジェクトの下に格納されます。 注意： スキャン プロセス中、指定されたプロジェクトが DTP データベース中にない場合、プロジェクトは自動的に作成されます。要求を適切に作成するために、qc-domain、qc-project、および dtp-project 属性が使用されます。

例：

```
<db-connection>
<db-type>MSSQL</db-type>
<user>sa</user>
<password encrypted="true">abc123</password>
<url>jdbc:sqlserver://localhost;</url>
<database qc-domain="DEFAULT" qc-project="QualityCenter_Demo"
dtp-project="MyProject">QualityCenter_Demo_db
</database>
</db-connection>
```

3. ota-connection を設定します。

Development Testing Platform のデータで QC の欠陥を自動的に更新できるようにするには、OTA API を使ったやり取りを可能にする証明書を指定する必要があります。それには、以下の一般的な設定を行います。

設定	説明
<bts>/<ota-connection>/<user>	OTA API を使ってやり取りできるユーザーの名前。
<bts>/<ota-connection>/<password>	指定されたユーザーのための有効なパスワード。

設定	説明
<bts>/<ota-connection>/ <url>	QC への有効な HTTP URL。

例：

```
<ota-connection>  
<user>sa</user>  
<password>sa</password>  
<url>http://localhost:8080/qcbin</url>  
</ota-connection>
```


Bugzilla との統合

Bugzilla との統合は、以下のバージョンでテスト済みです。

- 2.16
- 2.18
- 2.20
- 2.22
- 3.0
- 3.4
- 3.6
- 4.0
- 4.2

BTS Updater は Bugzilla 3.4 以降で動作します。

ビルトイン UI を使って Bugzilla と統合する方法については 277 ページの「BTS Updater の構成」を参照してください。

Bugzilla のスキャンを設定するには、DTP\grs\config\bts\bugzila.xml ファイルで以下の設定をカスタマイズする必要があります。

1. 以下の全般的な設定を指定します。

設定	説明
<bts>/<name>	Bugzilla サーバーを識別するためにユーザーが見て分かる名前を指定します (たとえば My Bugzilla Server)。Bugzilla サーバーの各インスタンスに固有の名前でなければなりません。この名前は Project Center の [欠陥 / 改善検索] ページで [欠陥トラッキング システム] ドロップダウン リストの選択肢として表示されます。
<bts>/<url-prefix>	Report Center サーバーから Bugzilla サーバーへのバグ詳細リンクを作成するために必要な接頭辞を指定します。
<bts>/<version>	DTP に統合する Bugzilla のバージョンを指定します。

2. 以下の Bugzilla データベース接続設定を指定します。

設定	説明
<bts>/<db-connection>/<db-type>	データベース ベンダー。MySQL または Oracle を指定します。
<bts>/<db-connection>/<user>	データベース ユーザー名

設定	説明
<bts>/<db-connection>/ <pass>	データベース パスワード
<bts>/<db-connection>/ <url>	Bugzilla データベース接続 URL

例 :

```
<bts type="Bugzilla">
  <name>My Bugzilla Server</name>
  <url-prefix>http://mybugzillaserver/bugzilla/
show_bug.cgi?id=</url-prefix>
  <version>3.4</version>
  <db-connection>
    <db-type>MySQL</db-type>
    <user>bugs</user>
    <password encrypted="true">abc123</password>
    <url>jdbc:mysql://bugzilla.host.com:3306/bugs</url>
<!-- sample Oracle connection url: <url>jdbc:oracle:thin:@HOST:PORT:SID/</
url> -->
  <!-- sample MySQL connection url: <url>jdbc:mysql://HOST:PORT/DATABASE/</
url> -->
  </db-connection>
</bts>
```

3. 「解決済みステータス」および「非アクティブの解決」の値を指定します。

言い換えると、バグが修正されたことを表すステータスと、バグがアクティブでないことを表すステータスを指定します。これらの情報がないと、Report Center はバグ履歴レポートを作成できません。

例 :

```
<resolved-status>
  <status>VERIFIED</status>
  <status>RESOLVED</status>
  <status>CLOSED</status>
</resolved-status>

<inactive-resolution>
  <resolution>INVALID</resolution>
  <resolution>WONTFIX</resolution>
  <resolution>DUPLICATE</resolution>
  <resolution>MOVED</resolution>
  <resolution>REMIND</resolution>
  <resolution>LATER</resolution>
  <resolution>WORKSFORME</resolution>
</inactive-resolution>
```

4. <feature-request> 設定を使って、バグが改善であることを表す、Bugzilla の重要度フィールドの値を指定します。改善を表す重要度の値は Bugzilla のバージョンによって異なります。

Bugzilla バージョン 3.0 以前の場合、改善を表す重要度の値は "Feature_Req" です。構成は次のようになります。

```
<feature-request>
  <item>Feature_Req</item>
</feature-request>
```

Bugzilla バージョン 3.4 の場合、改善を表す重要度の値は "enhancement" です。構成は次のようになります。

```
<feature-request>
  <item>enhancement</item>
</feature-request>
```

IBM Rational ClearQuest との統合

ClearQuest の統合は、以下のバージョンでテスト済みです。

- 2003.06.15.734.000
- 2007
- 7.0.1.1

ClearQuest と DTP を統合する前に、DTP がインストールされているのと同じマシンに ClearQuest クライアント プログラム (cqperl コマンド) がインストールされていることを確認してください。BTS Scanner は cqperl コマンドを使って ClearQuest から項目を読み込みます。

以下のディレクトリにある ExampleClearQuestScannerConfig.xml ファイルのコピーを作成します。

```
{DTP_HOME}\grs\config\bts\examples\
```

ClearQuest のスキャンを設定するには、コピーしたファイルに対して以下の設定をカスタマイズする必要があります。

1. 以下の全般的な設定を指定します。

設定	説明
<bts>/<name>	ClearQuest サーバーを識別するためにユーザーが見て分かる名前を指定します (たとえば My ClearQuest Defects Server)。ClearQuest サーバーの各インスタンスに固有の名前でなければなりません。この名前は Project Center の [欠陥 / 改善検索] ページで [欠陥トラッキング システム] ドロップダウン リストの選択肢として表示されます。
<bts>/<connection-settings>/<dbset>	ClearQuest データベース セット。
<bts>/<connection-settings>/<dbname>	ClearQuest データベース名。
<bts>/<connection-settings>/<user>	ユーザー名。
<bts>/<connection-settings>/<pass>	ユーザー パスワード。

2. <run-options> を使って、BTS Scanner で読み込む ClearQuest エンティティ名を定義します。

設定	説明
<run-options>/<entity-type>	インポート対象の項目（欠陥、変更リクエストなど）に関する情報を保持する ClearQuest データベースのエンティティ名。BTS Scanner はこのエンティティで指定された項目だけをスキャンします。

注意： ClearQuest からインポートしたい <entity-type> ごとに個別に BTS Scanner 構成ファイルを用意する必要があります。

3. (オプション) ClearQuest のエンティティの一部またはすべてを、DTP で改善として扱うかどうかを指定します。

ClearQuest のエンティティの一部またはすべてを、欠陥または改善のどちらとしてインポートするかを指定できます。デフォルトでは、ClearQuest の項目は欠陥として扱われます。

BTS Scanner は、インポートされたエンティティの severity フィールドの値に基づいてエンティティを改善として扱うかどうかを決定します。

ClearQuest のエンティティの一部またはすべてを改善として扱う場合、次の操作を行います。

- 「サンプル構成ファイルの準備」および「構成ファイルの追加」の説明に従って BTS Scanner 構成ファイルを準備します。
- 改善としてインポートするエンティティタイプのために <entity-type> を定義し、Clear Quest でのエンティティ名を指定します。

たとえば、ClearQuest では変更リクエストが "cr" エンティティとして保存されている場合、次のように <entity-type> を定義します。

```
<run-options>
  <entity-type>cr</entity-type>
</run-options>
```

- 以下のどちらかの操作を行います。

ClearQuest のエンティティの一部を改善として扱う場合：

- <feature-request> タグを使って、ClearQuest から読み取られたどの項目を改善として扱うかを指定します。

例：

```
<feature-request>/<item> 6-Enhancement </feature-request>
```

上記のように構成すると、Severity フィールドの値が "6-Enhancement" の項目はすべて改善として DTP にインポートされ、他の項目は欠陥としてインポートされます。

すべての ClearQuest エンティティを改善として扱うには、次のように構成します。

- <feature-request> タグに、取り得るすべての Severity フィールド値を指定します。

例：

```
<feature-request>
  <item>1-Critical</item>
  <item>2-High Attention</item>
  <item>3 -Major</item>
  <item>4-Average</item>
  <item>5-Minor</item>
  <item>6-Enhancement</item>
</feature-request>
```

4. ClearQuest データベース フィールドマッピングを指定します。

BTS Scanner は、Report Center のフィールドにマッピングされた ClearQuest のバグフィールドだけをインポートします。マッピングされていないフィールドは不明なフィールドとして扱われ、変換プロセス中にスキップされます。

注意：どのテーブルフィールドが自分の ClearQuest データベース中にあるか分からない場合、`read_single_bug.pl` スクリプトを実行して、ClearQuest データベースの 1 つのサンプル項目のテーブルフィールドを出力できます。このスクリプトは `DTP\grs\extras\ClearQuest` にあります。

このスクリプトを実行する前に、スクリプト ファイルを開いてログイン情報とサンプルの欠陥 ID を編集してください。

たとえば、ClearQuest の Headline フィールドは Report Center の Summary フィールドに対応し、Submitter フィールドは Reporter フィールドに対応するといったマッピングを指定します。

一部のフィールドは ClearQuest との統合に必ず必要ですが、他のフィールドはオプションです。

以下のフィールドマッピング設定が必要です。

フィールド	説明
<code><bts>/<fields-mapping>/<summary></code>	バグの概要。
<code><bts>/<fields-mapping>/<status></code>	現在のステータス。open、closed など。
<code><bts>/<fields-mapping>/<resolution></code>	バグの解決方法。fixed、wontfix など。
<code><bts>/<fields-mapping>/<priority></code>	バグの優先度。low、medium など。
<code><bts>/<fields-mapping>/<severity></code>	バグの重要度。minor、critical など。
<code><bts>/<fields-mapping>/<reporter></code>	バグの報告者。
<code><bts>/<fields-mapping>/<creation-date></code>	バグの作成日。

以下のフィールドマッピング設定はオプションです。

フィールド	説明
<bts>/<fields-mapping>/ <project>	バグが属するプロジェクト。
<bts>/<fields-mapping>/ <version>	プロジェクトのバージョン。
<bts>/<fields-mapping>/ <milestone>	プロジェクトのマイルストーン名。
<bts>/<fields-mapping>/ <hardware>	バグの影響を受けるハードウェア。
<bts>/<fields-mapping>/ <os>	バグの影響を受ける OS。
<bts>/<fields-mapping>/ <component>	プロジェクト コンポーネント。
<bts>/<fields-mapping>/ <modification-date>	バグの最終更新日。
<bts>/<fields-mapping>/ <assigned-to>	現在バグが割り当てられているユーザー。

例：

```
<fields-mapping>
  <summary>Headline</summary> *
  <status>State</status> *
  <resolution>Resolution</resolution> *
  <priority>Priority</priority> *
  <severity>Severity</severity> *
  <reporter>Submitter</reporter> *
  <creation-date>Submit_Date</creation-date> *
  <project>Project</project>
  <version>Version</version>
  <milestone>Milestone</milestone>
  <hardware>Hardware</hardware>
  <os>OS</os>
  <component>Component</component>
  <modification-date>Modification_Date</modification-date>
  <assigned-to>Owner</assigned-to>
</fields-mapping>
```

注意：アスタリスクが付いている項目は必須です。必須ではないフィールドを指定した場合、それらのフィールドの値に基づいて Report Center のプロジェクトにフィルターを追加することができます。

5. 「解決済みステータス」 および 「非アクティブの解決」 の値を指定します。

言い換えると、バグが修正されたことを表すステータスと、バグがアクティブでないことを

表すステータスを指定します。これらの情報がないと、Report Center はバグ履歴レポートを作成できません。

例：

```
<resolved-status>
  <status>Resolved</status>
  <status>Closed</status>
</resolved-status>

<inactive-resolution>
  <resolution>Duplicate</resolution>
  <resolution>Functions as Designed</resolution>
</inactive-resolution>
```

6. 以下の構成も必要です。

- <date-formats>

ClearQuest で使われている日付書式が以下のいずれかであることを確認してください。以下のリストにない場合、書式を追加してください。この情報は、DTP が ClearQuest から項目を読み込むときに、正しく日付を解釈するために必要です。

デフォルトの構成は次のとおりです。

```
<date-formats>
  <date-format>MMM-dd-yyyy HH:mm</date-format>
  <date-format>MM/dd/yyyy h:mm a</date-format>
  <date-format>yyyy-MM-dd HH:mm:ss</date-format>
</date-formats>
```

注意：書式は Java の SimpleDateFormat クラスに定義された書式のいずれかでなければなりません (<http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html> を参照)。

- <bugs-to-read>

この設定は、テスト目的でのみ使用されます。設定をテストするときに、何個の項目をインポートしたら BTS Scanner が停止するかを指定できます。たとえば 10 を指定すると、BTS Scanner は 10 個の項目をインポートした後に停止します。

注意：0 を指定すると、すべての項目が ClearQuest から読み込まれます。

- <retries-on-error>

まれに BTS Scanner からの cqperl の呼び出しが ClearQuest への接続に失敗する場合があります。この設定を指定すると、BTS Scanner は接続の問題があったときに cqperl の呼び出しを再試行します。

- <scan-changed-from>, <scan-changed-to>

この設定を指定すると、指定した期間に変更された項目だけがスキャンされます。

Atlassian JIRA との統合

JIRA との統合は以下のバージョンでテスト済みです。

- 3.6.2
- 3.7.3
- 3.10.1
- 3.10.2
- 3.12
- 3.13
- 4.1.1
- 4.1.2
- 4.3
- 4.4
- 5.x
- 6.x.

BTS Scanner と JIRA の連携を構成するには、JIRA 側と DTP 側の双方でいくつかの設定を行う必要があります。必要な調整について以下のセクションで説明します。

- JIRA サーバー側での設定
- DTP 側での JIRA の設定

BTS Scanner は HTTPS/SSL 経由で JIRA と連携するよう構成することもできます。詳細な方法については次のセクションを参照してください。

- HTTPS/SSL 経由で BTS Scanner と JIRA を連携するための構成

5.x および 6.x との統合

JIRA 5.x および 6.x との統合にはビルトイン UI を使用します (266 ページの「ビルトイン UI を使った BTS/RMS Scanner の構成」を参照)。JIRA 5.x および 6.x と統合するには、DTP Server で SSL を有効化する必要があります。詳細については 758 ページの「SSL を使用した Development Testing Platform の実行」を参照してください。

JIRA 4.4 以前との統合

以下のセクションでは、JIRA 4.4 以前の設定について説明します。ビルトイン UI を使って JIRA 4.x に対して DTP を設定できますが、JIRA サーバー側でも設定が必要です。このセクションの説明を参照してください。

すでに JIRA 4.x scanner を設定済みである場合 (たとえば 4.9.3 より前の DTP で JIRA 4.x scanner を設定済みである場合)、DTP の UI からさらに JIRA 5.x scanner を設定することはできません。これは制限事項ではなく、DTP で JIRA の課題が重複するのを防ぐための制約です。

既存の JIRA 4.x scanner を JIRA 5.x scanner に移行したい場合は Parasoft Support に連絡してください。移行を可能にするためにデータベースを手動で更新する必要があります。

JIRA サーバー側での設定

JIRA を設定する前に、JIRA サーバーで SOA インターフェイスが有効化されていることを確認する必要があります。JIRA リモート インターフェイスの有効化については、JIRA のドキュメントの RPC プラグインの有効化についてのトピックを参照してください。

<https://confluence.atlassian.com/display/ALLDOC/Atlassian+Documentation>

1. JIRA_INSTALL_DIR/atlassian-jira/WEB-INF/lib ディレクトリにある `atlassian-jira-rpc-plugin.jar` の名前を `atlassian-jira-rpc-plugin.jar.orig.bak` に変更して、既存のプラグインをバックアップします。
2. DTP に同梱された拡張された jar ファイルで、元のファイルを置き換えます。
[DTP_HOME]\grs\extras\jira ディレクトリの下に、JIRA のバージョンごとに jar ファイルが置かれています。

バージョン	Jar
3.6.*	[DTP_HOME]\grs\extras\jira\rpc-plugin.jar
3.7.3	[DTP_HOME]\grs\extras\jira\atlassian-jira-rpc-plugin-3.12.1-1.jar
3.10	[DTP_HOME]\grs\extras\jira\atlassian-jira-rpc-plugin-3.10.2-1.jar
3.12	[DTP_HOME]\grs\extras\jira\atlassian-jira-rpc-plugin-3.12.1-1.jar
3.13	[DTP_HOME]\grs\extras\jira\atlassian-jira-rpc-plugin-3.13.1-1.jar
4.1.2	[DTP_HOME]\grs\extras\jira\atlassian-jira-rpc-plugin-4.1.2.jar
4.3	[DTP_HOME]\grs\extras\jira\atlassian-jira-rpc-plugin-4.3.jar
4.4	[DTP_HOME]\grs\extras\jira\atlassian-jira-rpc-plugin-4.4.jar

JIRA 5.x との統合の詳細については 217 ページの「BTS Scanner と RMS Scanner の設定」を参照してください。

3. JIRA サーバーを再起動します。

DTP に同梱の拡張 JIRA SOA

DTP に同梱された JIRA SOA インターフェイスには、欠陥の履歴を引き出すメソッドが追加されています。このメソッドは [DTP_HOME]\grs\extras\jira ディレクトリにあります。

このメソッドの詳細については JIRA の Agile Board に掲載されています。

<http://jira.atlassian.com/browse/JRA-10333>

JIRA_INSTALL_DIR/atlassian-jira/WEB-INF/lib/atlassian-jira-rpc-plugin.jar には、JIRA への SOA アクセスの実装が含まれています。追加された SOA メソッドについて JIRA の管理者に通知することを推奨します。

DTP は他の JIRA SOA メソッドを使用します。オリジナルの JIRA SOA メソッドは変更されません。そのため、jar ファイルを置き換えても SOA インターフェイスが拡張されるだけであり、JIRA でデグレードが起こることはありません。

この拡張は通常の JIRA ディストリビューションには含まれていませんが、Atlassian の初期承認を受けています。

DTP 側での JIRA の設定

1. JIRA のスキャンを設定するには、[DTP_HOME]\grs\config\bts\Jira.xml ファイルで以下の全般的な設定を行います。

設定	説明
<bts>/<name>	JIRA サーバーを識別するためにユーザーが見て分かる名前を指定します (たとえば My JIRA Server)。JIRA サーバーの各インスタンスに固有の名前でなければなりません。この名前は Project Center の [欠陥 / 改善検索] ページで [欠陥トラッキング システム] ドロップダウン リストの選択肢として表示されます。
<bts>/<url-prefix>	Report Center から JIRA サーバーのバグ詳細へのリンクを作成するために必要な接頭辞。
<bts>/<connection-settings>/<soap-service>	JIRA SOAP サービスのアドレス。通常は次の書式です。 http://<YOUR_JIRA_SERVER_ADDRESS>:<SERVER_PORT>/rpc/soap/jirasoapervice-v2</soap-service>
<bts>/<connection-settings>/<pass>	JIRA ユーザー パスワード。
<bts>/<connection-settings>/<timeout>	BTS Scanner/JIRA の接続タイムアウトをミリ秒単位で指定します。指定されていない場合、Axis のデフォルトのタイムアウトが使用されます。

例 :

```
<connection-settings>
  <soap-service>http://localhost:8080/rpc/soap/jirasoapervice-v2
</soap-service>
  <user>root</user>
  <pass>root</pass>
  <timeout>720000</timeout>
</connection-settings>
```

2. 「解決済みステータス」および「非アクティブの解決」の値を指定します。この情報がないと、Report Center はバグ履歴レポートを作成できません。

例 :

```
<resolved-status>
  <status>Resolved</status>
  <status>Closed</status>
</resolved-status>
```

```

<inactive-resolution>
  <resolution>Duplicate</resolution>
  <resolution>Won't Fix</resolution>
</inactive-resolution>

```

3. (オプション) JIRA の課題をスキャンするためのスコープを指定します。読み込むことができるのは、「JIRA のすべての課題」、「プロジェクトで制限された課題 (JIRA プロジェクトキーで指定)」、および「ユーザー定義の JIRA フィルターで定義された課題 (JIRA フィルター ID で指定)」です。課題を読み込むときにフィルターのためにページ分割を利用できません。たとえば、JIRA へのリクエスト (SOAP 呼び出し) のたびに 1000 件の課題を読み込むようにスキャナーをセットアップできます。1 つの JIRA リクエストにおいて、スキャン対象の項目の数が多すぎる場合 (たとえば 10000 個以上の場合)、ページ分割を利用してメモリの問題を避けることができます。

例:

```

<scanner-scope>
  <!--
    Scan Jira items which belong to the following Jira filters
  -->
  <jira-filters enable-paging="true" items-per-page="1000">
    <filter-id>10035</filter-id>
    <filter-id>10036</filter-id>
  </jira-filters>

  <!--
    Scan the following Jira projects (project keys)
    If <jira-filters> is specified, this <jira-projects> section is skipped
  -->
  <jira-projects>
    <project-key>PROJECTONE</project-key>
    <project-key>PROJECTTWO</project-key>
  </jira-projects>
</scanner-scope>

```

- `enable-paging`
読み込みのページ分割が有効 (`true`) か無効 (`false`) かを表します。
- `items-per-page`
一度に何件の課題を読み込むかを指定します (1000 を指定すると、JIRA リクエストごとに 1000 件の課題が読み込まれます)。
- `<filter-id>`
JIRA で作成されたフィルターの ID を指定します。JIRA で作成されるフィルター定義は、BTS Scanner 構成を JIRA に接続するユーザーが参照できるべきです。JIRA でこのフィルターが作成された後、ユーザーは JIRA からフィルター ID を取得できます。まず、JIRA の Web ページにアクセスし、[Issues] > [Manage Filters] を選択します。[Manage Filters] ページで、フィルターの [edit] リンクをクリックします。次に、このページの URL を見て `filterId` パラメーターを確認します。たとえば次の URL の場合、`filterId` は 10032 です。

```
http://xen3.parasoft.com.pl:8080/secure/
EditFilter!default.jspa?atl_token=a08P-
8w_fk&filterId=10032&returnUrl=ManageFilters.jspa
```

まだそうでない場合、JIRA の他のユーザーとフィルターを共有してください。

4. (オプション) JIRA のカスタム タイプ マッピングを指定します。

デフォルトでは、JIRA の項目をインポートするとき、DTP は JIRA の "New Feature" および "Improvement" タイプを DTP の「フィーチャー リクエスト」として、そして JIRA の "bug" タイプを DTP の「バグ」として扱います。これは XML の以下の部分であらかじめ定義されています。

```
<imported-issue-type>
  <issue-type translate-to="BUG">Bug</issue-type>
  <issue-type translate-to="FEATURE_REQUEST">New Feature</issue-type>
</imported-issue-type>
```

- JIRA でカスタム タイプを定義している場合、このノードを使ってカスタム タイプをインポートし、DTP のバグまたはフィーチャー リクエストにマッピングできます。たとえば、"My Defect" として JIRA のカスタム タイプを定義している場合、以下のように DTP バグにマッピングできます。

```
  <imported-issue-type>
    <issue-type translate-to="BUG">My Defect</issue-type>
    <issue-type translate-to="BUG">Bug</issue-type>
    <issue-type translate-to="FEATURE_REQUEST">New Feature</issue-type>
    <issue-type translate-to="FEATURE_REQUEST">Improvement</issue-type>
  </imported-issue-type>
```

- config xml 中に <imported-issue-type> ノードがない場合、下位互換性のためにデフォルトの構成が使用されます。
- <imported-issue-type> ノードが空の場合、JIRA の項目はインポートされません。

HTTPS/SSL 経由で BTS Scanner と JIRA を連携するための構成

HTTPS/SSL 経由で BTS Scanner と JIRA を連携するには、通常と同じように JIRA 用に BTS Scanner を構成した後で、以下で説明する手順を実行します。

- BTS Scanner 構成を変更する
- キーストア パラメーターを指定して DTP サービスを実行する

BTS Scanner 構成を変更する

SOAP サービス アドレスに以下の変更を加えます。

1. http を https に変更します。
2. 適切な SSL ポート (通常は 443 または 8443) を指定します。

例：

```
<connection-settings>
  <soap-service>https://jira.somecompany.com:8443/rpc/
soap/jirasoapservice-v2</soap-service>
  <user>root</user>
  <pass>root</pass>
</connection-settings>
```

キーストア パラメーターを指定して DTP サービスを実行する

1. JIRA tomcat の使用する SSL キー ファイルを取得します。

必要に応じて、この手順について以下で詳細を参照できます。

```
http://confluence.atlassian.com/display/JIRA/
Running+JIRA+over+SSL+or+HTTPS (JIRA ドキュメント)
```

および

```
http://java.sun.com/javase/6/docs/technotes/guides/security/
jsse/JSSERefGuide.html#CreateKeystore
```

2. キー ファイルを DTP ホストの任意のディレクトリに保存します。
3. DTP Server に `Java -Djavax.net.ssl.trustStore=<TRUSTED KEYSTORE PATH>` パラメーターを設定し、SSL キー ファイルの場所を指定します。

次の例では、キー ファイル名が `.keystore` であると仮定しています。

Linux の場合、たとえばキー ファイルを次の場所に保存したと仮定します。

```
/home/ser/.keystore
```

1. `DTP_HOME/bin/reportserver.sh` を編集します。

2. 変更前：

```
export CATALINA_OPTS="-D$PCC_RECOGNITION \
```

変更後：

```
export CATALINA_OPTS="-D$PCC_RECOGNITION -Djavax.net.ssl.trustStore=/
home/ser/.keystore \
```

Windows の場合、たとえばキー ファイルを次の場所に保存したと仮定します。

```
D:\home\user\dtp\.keystore
```

1. `regedit` コマンドを実行して Windows レジストリ エディターを開きます。
2. Parasoft DTP サービスの呼び出しパラメーターを探します。

通常、次の場所にあります。

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Parasoft DTP
Server Service
```

3. 新しい JVM Option Number パラメーターを追加します。

たとえば、番号が 0 から 14 までの 15 個のパラメーターがある場合、次のように 16 個目のパラメーターを追加します。

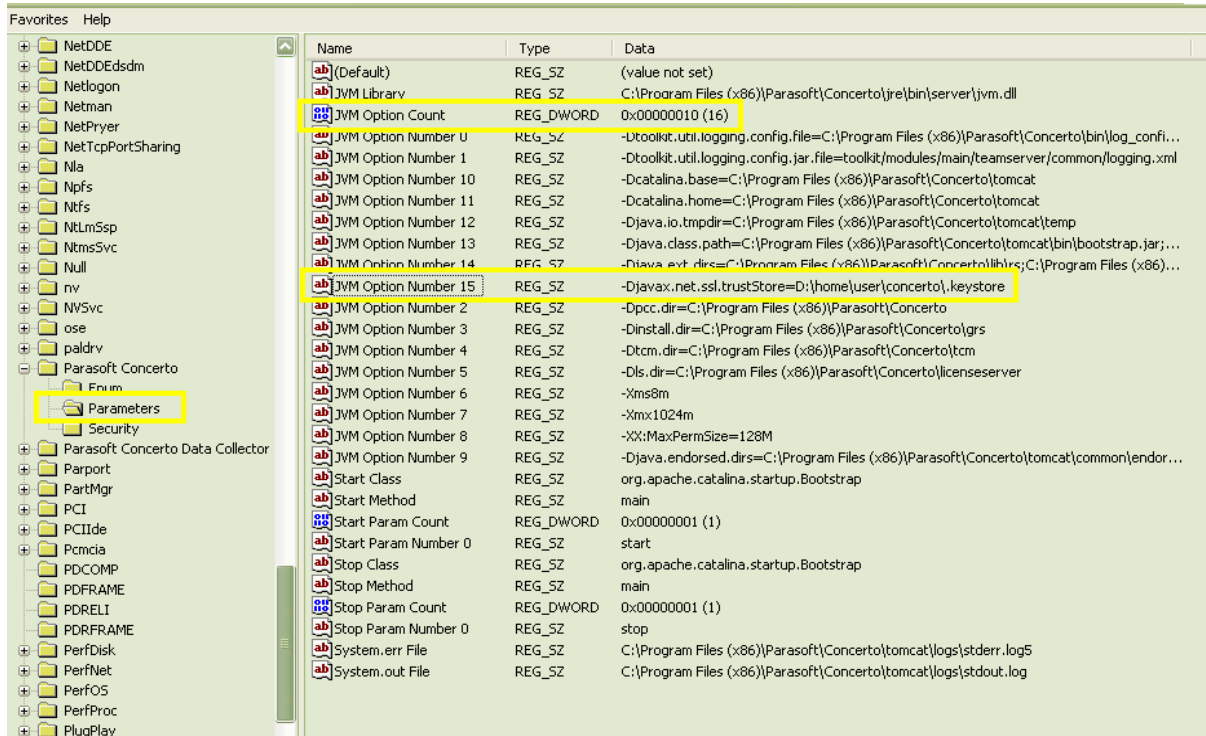
JVM Option Number 15

値：

-Djavax.net.ssl.trustStore=D:\home\user\ntp\keystore

4. パラメーターのカウンター JVM Option Count の値を増やします。たとえば、JVM Option Count の値が 15 だった場合、16 にします。

Regedit での JVM Option Count の編集



HTTP ベーシック認証を使用した BTS Scanner と JIRA の連携

HTTP ベーシック認証を使用して BTS Scanner と JIRA を連携するには、通常どおり JIRA 用に BTS Scanner を構成した後で、以下の手順を実行します。

`${DTP_HOME}\grs\config\bts\Jira.xml` ファイルの接続設定に `<basic-http-authentication>` を追加します。

```
<basic-http-authentication>
  <user>admin</user>
  <pass encrypted="false">admin</pass>
</basic-http-authentication>
```

例：

```
<connection-settings>
  <soap-service>https://jira.somecompany.com:8443/rpc/soap/
jirasoapservice-v2</soap-service>
  <user>root</user>
  <pass>root</pass>
  <basic-http-authentication>
    <user>admin</user>
    <pass encrypted="false">admin</pass>
  </basic-http-authentication>
</connection-settings>
```

注意： <basic-http-authentication> タグ内のユーザーおよびパスワードが HTTP ヘッダーのパラメーターとして JIRA サーバーに送信されます。

IBM Rational Change および Rational Synergy との統合

このセクションでは、Rational Change の成果物を Development Testing Platform (DTP) にインポートする方法と DTP にインポートされた Rational Change の成果物を使用するようにタスクアシスタントを設定する方法について説明します (310 ページの「IBM Rational Change および Rational Synergy との統合」)。

Rational Change との統合

Rational Change と Rational Synergy 用の BTS Scanner を使用すると、変更リクエスト (および関連するタスク) を Rational Change から DTP にインポートすることができます。Rational Change の変更リクエストは DTP では要求になります。Rational Change と Rational Synergy は BTS/RMS Scanner によって同期化されます。BTS/RMS Scanner は自動的に実行するか、必要に応じて手動で実行できます。

1. DTP_HOME/grs/config/bts/examples/ExampleSynergyScannerConfig.xml ファイルのコピーを作成します (このファイルには Synergy の設定のサンプルが含まれます)。実際の Synergy サーバーと DTP Server に合わせて、コピーしたファイルを編集します。個々の設定要素がこの XML ファイル中に記述されています。
2. このファイルを DTP_HOME/grs/config/bts にコピーします。ここまでの操作が終了すると、指定した Rational Change サーバーが接続され、.xml ファイルで指定された "ccm" クエリーが 15 分ごとに実行されます (または、ユーザーが Project Center の [要求] ページで [再スキャン] ボタンをクリックしたときに実行されます)。

クエリーによって発見された変更要求 (およびその変更要求に関連付けられた子要求とタスク) は、DTP のターゲット プロジェクト中の要求およびタスクと同期化されます。

Rational Change で行われる変更は DTP で次のように更新されます。

- 特定の要求またはタスクが DTP に存在しない場合、その要求またはタスクは追加されます。
- 特定の要求またはタスクが Rational Change で変更された場合、その要求またはタスクは DTP で更新されます。Rational Change の変更リクエスト ID とタスク ID が DTP の要求のオリジナル ID とタスクのオリジナル ID フィールドにそれぞれ格納されます。

注意

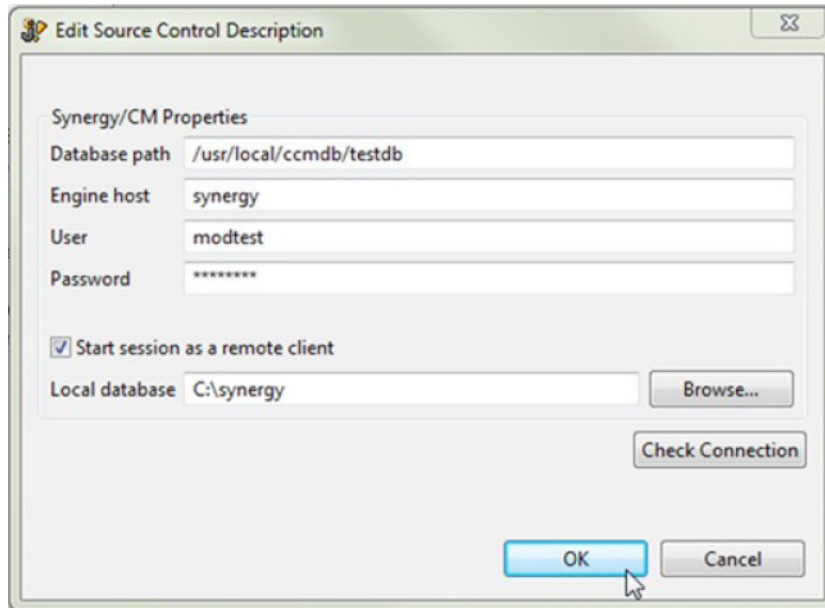
- .xml 設定ファイルで設定された Synergy の "ccm" クエリーは、DTP ターゲット プロジェクトに緊密に統合されます。たとえば、特定の構成が返す変更リクエストを減らすために、クエリーを変更または削除した場合、DTP ターゲット プロジェクト中の未解決の要求が削除されます。削除された要求に関連付けられたすべてのタスクの種類が "スタンダードアロン" に変わります。タスクが削除されることはありません。
- DTP の要求オリジナル ID およびターゲット プロジェクトは、DTP/Rational Change で対応する項目を探すときに考慮されます。たとえば、ある要求がインポートされ、その要求と同じオリジナル ID を持つ要求が DTP ターゲット プロジェクトに存在しない場合、DTP はその要求を作成します。インポートされた要求と同じオリジナル ID を持つ要求が DTP ターゲット プロジェクトに存在する場合、その要求は更新されます。
- 場合によっては、DTP プロジェクトと同期化するように "ccm" クエリーを定義することもあります。そして、構成の DTP ターゲット プロジェクト名は変更しつつ、クエリーに関しては、別の DTP プロジェクトと同じ変更リクエストをインポートできるよう、そのままにすることもあります。その後、Rational Change の「まだ DTP にインポートされていない変更リクエスト」または「前回 DTP にインポートされてから修正があった変更リクエスト」を、"ccm" クエリーで検索して DTP にインポートする場合、オリジナル ID が同じ要求が DTP にすでに存在するが、その要求が別のプロジェクト中にあると、以下の処理が実行されます。
 - 要求はオリジナル プロジェクトから削除されます。
 - 削除される要求に関連付けられたすべてのタスクについて、その種類が "スタンダードアロン" に変わります。タスクは削除されません。
 - 要求は新しいプロジェクトにインポートされます。
 - このインポートされる要求にタスクが関連付けられている場合、オリジナル ID に基づいて、タスクは新規に作成されるか、またはこの要求に移動されます。

3. クエリーが検索する変更リクエストごとに、関連する子要求とタスクも同期化されます。たとえば、新しいタスクが Rational Change 中の変更リクエストに追加される場合、このタスクは同期化の際に DTP にも追加されます。

この時点で、DTP ターゲット プロジェクトで要求とタスクが見えるはずですが、DTP にインポートされた要求とタスクのオリジナル ID フィールドに、Rational Change での ID が表示されている点に注意してください。

DTP および Rational Change Synergy と連携するための Parasoft Test タスク アシスタントの設定

1. Synergy からデータをインポートした DTP プロジェクトを構成します。次の操作を行います。
 - a. Parasoft Test で、Synergy ソース リポジトリにアクセスするためのテキスト プロパティを取得します。それには、Parasoft Test で Synergy のソース管理を定義し、その設定をファイルにエクスポートします。



エクスポートしたプロパティを DTP プロジェクトの [Parasoft Test 設定] タブに貼り付けます。以下はエクスポートしたプロパティの例です。

```
scontrol.rep1.synergy.dbpath=/usr/local/ccmdb/testdb
scontrol.rep1.synergy.host=synergy
scontrol.rep1.synergy.local_dbpath=C:\\tmp\\local_synergy_db
scontrol.rep1.synergy.login=modtest
scontrol.rep1.synergy.password=6e6b703e6952746f
scontrol.rep1.synergy.remote_client=true
scontrol.rep1.type=synergy
scontrol.synergy.exec=ccm
```

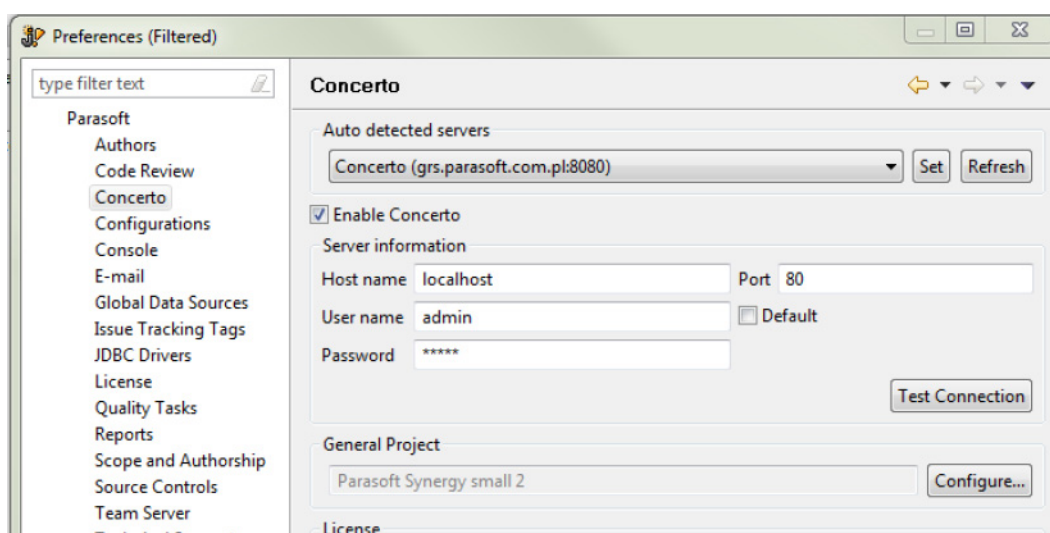
この設定は、指定の DTP プロジェクトに Parasoft Test が接続するときに使用されます。この設定はプロジェクト固有です。

- b. DTP の管理ページで、作業するプロジェクトを開いて編集し、[Parasoft Test 設定] タブをクリックします。上記のプロパティを貼り付けます。この設定をすべての DTP プロジェクトに対してグローバルに指定する場合は、プロジェクトの [Parasoft Test 設定] タブではなく、[Administration] > [Settings] > [Parasoft Test] で表示されるページでこの情報を貼り付けます。
- c. [Administration] > [Settings] > [Parasoft Test] ページに移動し、次の行を [Global Parasoft Test Settings] フィールドに貼り付けます。

```
tasksImportedMode=true
```

この設定は、この DTP Server に接続するタスク アシスタントを Synergy モードに切り替えます。つまり、タスク アシスタントはタスク ID であるかのようにタスク オリジナル ID を処理します (Synergy のタスク ID の値は、タスク オリジナル ID として DTP で表示されます)。

2. DTP Server ホストで、Synergy クライアントをインストールします。ccm コマンドを使って、DTP は Synergy のタスクと関連付けられたファイルを読み込みます (DTP は、タスク アシスタントが照会するときにこれらのファイルを読み込みます)。
3. 以下のことを確認してください。
 - DTP Server サービスから見えるシステム パス上に "ccm" コマンドがある。
 - たとえスペースを含む場合でも、定義されたパスがダブルクォートで囲まれていない。
4. Parasoft Test で DTP の設定を開き、DTP Server を指定します。そして [General Project] に、Synergy の要求とタスクをインポートした、構成したプロジェクトを指定します。



ここまでの操作によって、タスク アシスタントと Synergy を連携する準備ができました。つまり、DTP からタスクを検索するクエリーをユーザーは定義できます (そして DTP は Synergy からタスクを検索します)。特定のタスクに関連付けられたファイルが Synergy にある場合、タスク アシスタントでそのタスクをアクティブにすると、DTP はそれらのファイルのリストを Synergy から読み込んで タスク アシスタントに表示します。

トラブルシューティング

ccm プログラムを実行できない

Synergy タスクをアクティブにしようとする、Parasoft Test のタスク アシスタントで「Fail reason: Cannot run program ""ccm"": CreateProcess error=2」というエラーが出る。

考えられる原因：

- Synergy クライアント (ccm コマンドを含む) が DTP Server マシンにインストールされていない。
- DTP サービスに見えるシステム パス上に ccm コマンドがない。

- ccm.exe コマンドへのパスが Windows で定義されているが、二重引用符で囲まれて定義されている。

解決方法： Synergy クライアント (ccm コマンドを含む) が DTP Server マシンで使用できること、DTP サービスに見えるシステムパス上に ccm コマンドがあること、および Windows の ccm コマンドへのパスが (たとえパスにスペースが含まれる場合でも) 引用符で囲まれていないことを確認します。

新しい Synergy/CM セッションを開始できない

Synergy タスクをアクティブにし、関連するファイルを見るために [タスク アシスタント] ビューに移動すると、「Cannot start new Synergy/CM session. Warning: IBM Rational Synergy startup failed」というエラーが出る。

考えられる原因： DTP Server から ccm コマンドは見えているが、DTP サービスを実行しているシステムユーザーに Synergy セッションを開始する権限がない。

解決方法： Synergy セッションを開始する権限を持った Windows システムユーザーとして DTP サービスを開始します。Windows で DTP サービスを実行するユーザーを変更する方法については 355 ページの「ソース表示機能」の「解決方法 2」を参照してください。

デバッグ モードを有効にする

次の a) b) c) に該当する場合、DTP Server ログをデバッグ レベルに切り替えます。操作手順については 358 ページの「デバッグ ログ モードへの切り替え」を参照してください。

- a) Synergy の特定のタスクに関連付けられたファイルがあるが、それらのファイルをタスク アシスタントで見ることができない。
- b) それらのファイルを照会すると、タスク アシスタントでエラーが発生する。
- c) 上記のヒントが役に立たなかった。

コードレビューの統合

Development Testing Platform (DTP) は、Parasoft Test (SOAtest、Jtest など) を介してユーザーがセットアップしたコードレビューと統合します。その過程で Parasoft Test はコードレビューエンジンの役割を果たし、チームメンバーは Parasoft Test を介してコードレビュータスクを処理します。

DTP は、コードレビューの結果を格納するために使用され、コードレビューの結果についての洞察を提供します。

Parasoft Test でコードレビューを設定して実行する方法については『Parasoft Test ユーザーズガイド』を参照してください。

プロジェクトのコードレビューレポートの設定

コードレビューレポートは、選択されたプロジェクトについてコードレビューの統計を示します (コードレビューレポートについては 68 ページの「コードレビューレポート」を参照)。

コードレビューレポートを生成するときに DTP は次のことを考慮します。

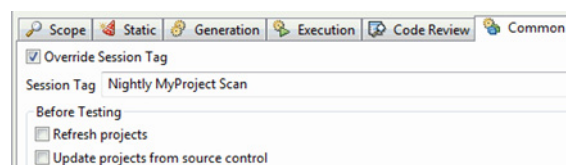
- そのプロジェクトのメンバーに関連付けられた結果はどれか。
- コードレビューセッションの識別子 / セッションタグでマークされている結果はどれか。

レポートを設定するには、次の操作を行います。

1. プロジェクトのチームメンバーシップのリストに、コードレビュープロセスに関わるユーザーが含まれていることを確認します。
 - チームメンバーの構成の詳細については 180 ページの「プロジェクトの作成と構成」を参照してください。
2. プロジェクトのコードレビューフィルターに、Parasoft Test Code Review テストコンフィギュレーションに一致するセッションタグセットがあることを確認します。
 - コードレビューフィルターの詳細については 186 ページの「コードレビューフィルター」を参照してください。

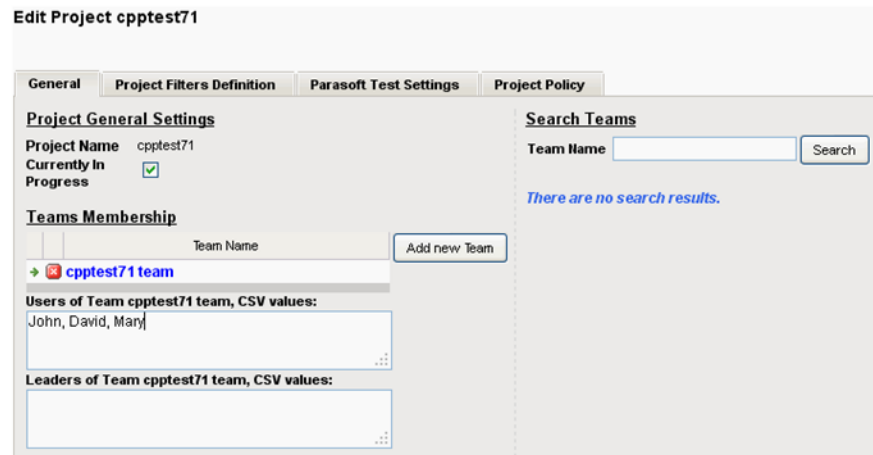
たとえば、次の状況を想定してみましょう。

- John、David、および Mary のコードレビューの結果を参照したい。
- チームの Code Review テストコンフィギュレーションが Nightly MyProject Scan セッションタグを使用している。

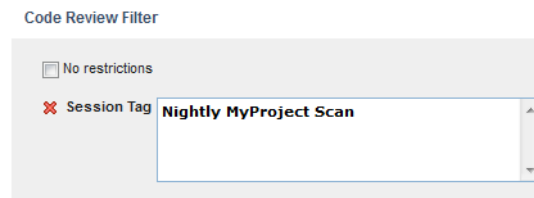


この場合、以下の設定を DTP で行います。

- チームメンバーシップのリストに John、David、Mary を追加します。



- Nightly MyProject Scan セッションタグを使うようにコードレビューフィルターを設定します。



コードレビューデータの格納

DTP でコードレビューデータを格納する方法は 2 つあります。

- Report Center データベースを使用する（推奨。より効率的にデータを格納できます）。
- Team Server を使用する（下位互換のためにサポートされている）。

コードレビューデータの格納方法によって、DTP の統合手順は異なります。

Report Center データベースを使用する場合

このオプションを推奨します。このオプションを選択した場合、Parasoft Test で行われたコードレビューの構成で十分です。DTP 側でさらに構成を行う必要はありません。

Report Center のレポートは、Report Center データベースからデータを取得するようにデフォルトで構成されます。

Team Server を使用する場合

Report Center のレポートは、デフォルトでは Report Center データベースからデータを取得するように設定されています。Team Server からデータを取得するように切り替えるには、次の操作を行います。

1. Report Center で、[管理] ドロップダウンメニューから [Report Center] を選択し、[設定] > [Report Center] > [サーバー] をクリックします。[非推奨の Team Server ベースのコード レビューを使用] チェックボックスをオンにします。
2. DTP_HOME/grs/config/CRHistoryScanners.xml ファイルを編集し、Team Server からデータを読み込んで Report Center データベースに保存するようにナイトリー ジョブを設定します。コード レビュー データを格納した Team Server の各サーバーについての情報を提供します。以下はその例です。

```
<configuration>
  <cr-history-scanner>
    <teamserver>
      <host>teamserver1.company.com</host>
      <port>1111</port>
      <login>login</login>
      <password>pass</password>
      <timeout>7200000</timeout>
    </teamserver>
  </cr-history-scanner-->
  <cr-history-scanner>
    <teamserver>
      <host>teamserver2.company.com</host>
      <port>1111</port>
      <login>login2</login>
      <password>pass2</password>
      <timeout>7200000</timeout>
    </teamserver>
  </teamserver>
</cr-history-scanner>
  ...
</configuration>
```

3. DTP Server を再起動して変更を反映します。

Report Center Code Review History Scanner ジョブが起動すると、<cr-history-scanner/> セクションに指定されたすべての Team Server が順番にスキャンされます。そしてデータが Report Center データベースに保存されます。

スキャン処理には時間がかかるため、このジョブはあらかじめ構成されているとおり、独立したグループとして実行するべきです。time-out パラメーターには、スキャン処理が完了するまでのタイムアウト時間をミリ秒単位で指定します。デフォルト値の 7,200,000 ミリ秒を設定するべきです。

警告！ Team Server リポジトリの初回のスキャンは非常に長い時間がかかる場合があります。これは、Report Center Code Review プラグインがインデックスをゼロから作成する必要があるからです。コード レビュー アクティビティの完全な履歴を取得するには、スキャン セッションを何度か実行する必要がある場合もあります。その後のスキャンにかかる時間は一日一回実行できる程度に短縮されるでしょう。

Report Center と Emma の統合

Report Center は、Java のコード カバレッジ ツールである Emma と統合できます。Emma の機能テスト カバレッジの結果が Report Center レポートに表示されます。319 ページの「Emma の結果の参照」を参照してください。

1. Report Center で、[管理] ドロップダウン メニューから [Report Center] を選択します。
2. [設定] メニューの [統合] > [Emma] をクリックします。
3. すべてのテスト対象クラスのカバレッジを含む Emma の .xml ファイルの URL を入力します。

Report Center は、この URL の場所を定期的にスキャンします。新しい結果が発見された場合、その結果は Report Center データベースにロードされます。

注意： 指定した URL にある .xml ファイルは HTTP サーバーによって提供されていなければなりません。 .xml 出力のサンプルを Emma の公式 Web サイトで参照できます。

http://emma.sourceforge.net/coverage_sample_c/coverage.txt

4. (オプション) すべてのテスト対象クラスのカバレッジを含む Emma の .html ファイルの URL を入力します。この HTML ファイルには、ファイルには Emma の最終実行時の結果が含まれていなければなりません。

UTL の場所は定期的にスキャンされません。DTP は、Report Center のカバレッジ レポートの [Emma Coverage] 列から Emma のカバレッジ レポートにリンクします。319 ページの「Emma の結果の参照」を参照してください。

指定した URL にある .html ファイルは HTTP サーバーによって提供されていなければなりません。Emma HTML レポートのサンプルを Emma の公式 Web サイトで参照できます。

http://emma.sourceforge.net/coverage_sample_a/index.html

5. [User Attribute: Project] フィールドで Test Group Property 属性フィルターを指定します。このフィルターは、どの DTP プロジェクトを Emma の結果と関連付けるかを指定します。フィルターの値は Edit Project ページで設定されます。詳細については 184 ページの「テスト グループ プロパティ フィルター」を参照してください。

PARASOFT Development Testing Platform Report Center

Main Projects Tools Settings Reports Help

Emma Integration Settings

The below Emma results file URIs can point to remote file (<http://example.com/coverage.xml>) or local file on DTP server (<file:///c:/emma/coverage.xml>)

Emma Scanner Configuration1

Emma results .xml file URI:

Emma results .html file URI:

User Attribute: Project

[Add Another Configuration]

Save

6. さらに設定を行うか、[Save] をクリックして操作を続けます。
7. DTP_HOME\grs\config\CronConfig.xml ファイルを編集し、次の xml ノードに移動します。

```

<Job allowFromHour="0" allowToHour="23" frequency="1440"
id="Emma Scanner"runDayOfWeek="*" runHour="2" runMinute="0">
    <class allowInSlave="false"
name="com.parasoft.grs.rserver.cronjobs.EmmaScannerJob"
priority="1"/>
</Job>

```

このノードには、ステップ 1 で定義された場所をスキャンする周期的な Report Center バックグラウンド ジョブが設定されています。この例では毎晩午前 2 時にジョブが実行されます。

必要に応じて実行時間を変更します。変更を行ったら、DTP Server を再起動して変更を有効にします。

8. Report Center のカバレッジ レポートを Emma モードに切り替えます。

a. 以下のファイルを編集します。

- DTP_HOME/grs/xreports/architect_dashboard/CoverageDetails.xml
- DTP_HOME/grs/xreports/architect_dashboard/CoverageOverview.xml
- DTP_HOME/grs/xreports/architect_dashboard/CoverageOverviewDetails.xml
- DTP_HOME/grs/xreports/architect_dashboard/composite_desc.xml
- DTP_HOME/grs/xreports/lite_dashboard/composite_desc.xml

b. 次の XML 要素に移動します。

- `<parameter name="show_emma_results">false</parameter>`

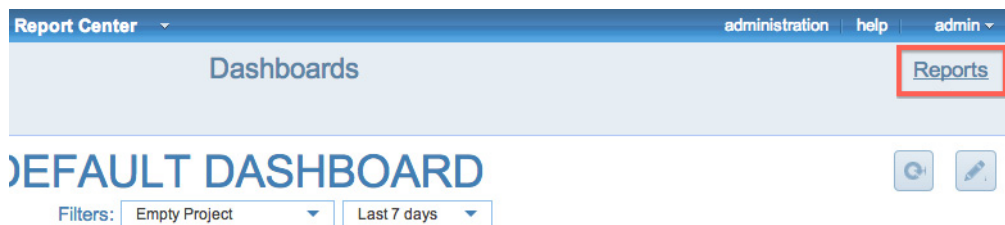
c. 値を true に設定します。

- `<parameter name="show_emma_results">true</parameter>`

Report Center と Emma の統合が完了します。

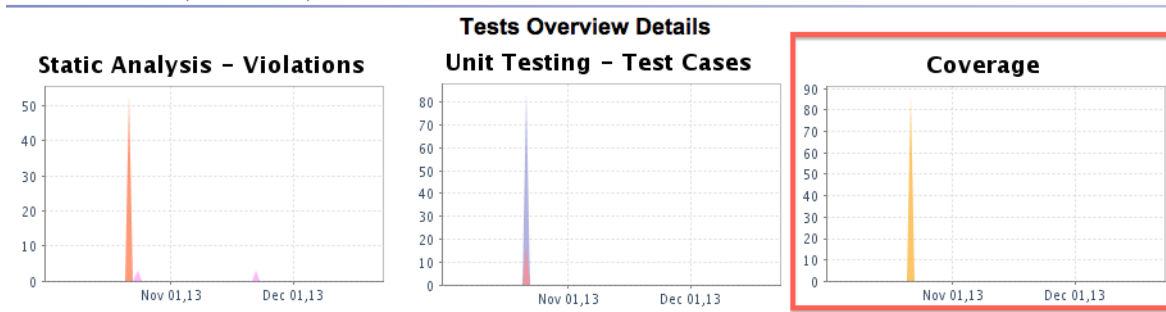
Emma の結果の参照

1. Development Testing Platform レポート ビューを開きます。



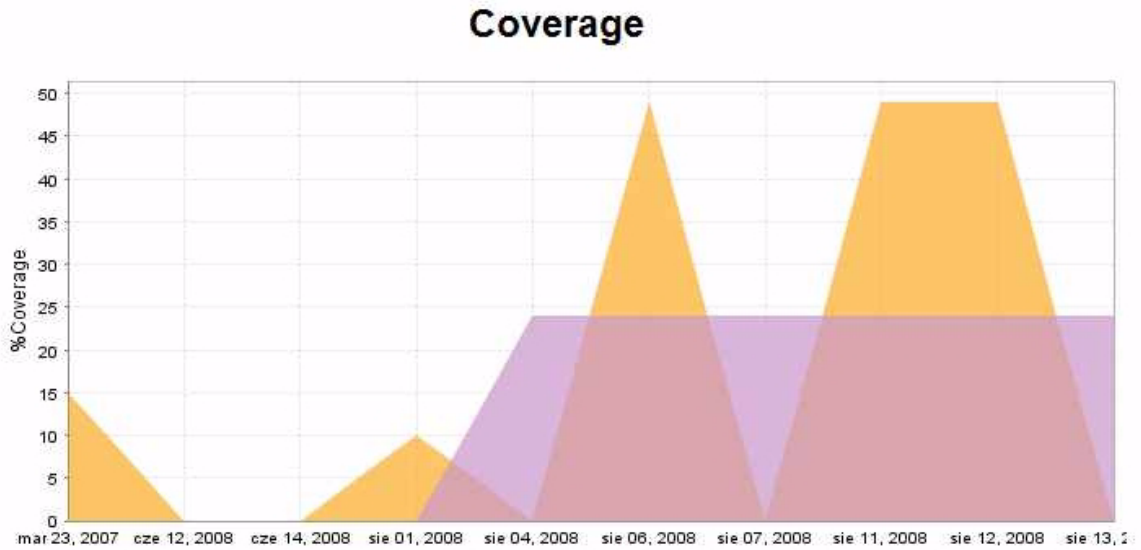
2. [テスト] > [テストの概要] をクリックします。

3. カバレッジ レポートをクリックします。



Emma モードでは、カバレッジ レポートとドリル ダウン レポートに [Emma Coverage] 列が表示され、Emma からレポートされたカバレッジが表示されます。

列のヘッダーをクリックして Emma HTML ファイルにアクセスします。



Date	Unit Tests Coverage	Emma Coverage
sie 13, 2008	0% (0 / 0)	24% (5466 / 23157)
sie 12, 2008	49% (1582 / 3256)	24% (5466 / 23157)
sie 11, 2008	49% (791 / 1628)	24% (5466 / 23157)
sie 07, 2008	0% (0 / 0)	24% (5466 / 23157)
sie 06, 2008	49% (791 / 1628)	24% (9110 / 38595)
sie 04, 2008	0% (0 / 0)	24% (1822 / 7719)
sie 01, 2008	10% (791 / 8035)	0% (0 / 0)
cze 14, 2008	0% (0 / 0)	0% (713 / 156813)
cze 12, 2008	0% (0 / 0)	0% (1426 / 313626)
mar 23, 2007	15% (26250 / 175526)	0% (0 / 0)

Report Center とソース管理システムの統合

Report Center とソース管理システム (SCM) を統合すると、コミットされたソース ファイルからのデータを基に Report Center で広範な統計を表示することができます。

DTP を SCM と統合すると、ソース リポジトリに保存されたファイルを SourceScanner (SDLC Extensions に同梱) がスキャンし、Report Center データベースにリビジョン情報を格納します。Report Center が収集するデータを使用して正確なレポートが生成されます。Report Center は、ソース コードに関連するレポートに、適切なリポジトリのソース コードを表示できます。

DTP を SCM に統合しない場合は、DTP Engine から DTP Server に送られたソース ファイルの内容を参照することができます。詳細については 644 ページの「DTP Engine がパブリッシュしたソース コードの表示」を参照してください。

Parasoft SDLC Extensions

SourceScanner は Development Testing Platform には同梱されていません。SourceScanner は Parasoft SDLC Extensions モジュールの一部です。

SourceScanner の構成

SourceScanner を構成して、SourceScanner から Report Center にデータを送信します。

SourceScanner は他の SDLC integration extensions とともに提供されています。

SourceScanner の構成方法については、『Parasoft SDLC Integration Extensions ユーザーズ ガイド』を参照してください。

Report Center プロジェクトに対するソース管理フィルターの定義

SourceScanner から Report Center にデータを送信できるように構成したら、特定のプロジェクトのレポートに表示させたいリポジトリ ファイルを定義する必要があります。詳細な手順については 185 ページの「ソース管理フィルター」を参照してください。

重要! ソース リポジトリの特定の場所をスキャンする場合、1 つの SourceScanner プロジェクトだけを使用することを強く推奨します。つまり、2 つのブランチおよび同じファイルのトランクをスキャンする場合、3 つの個別の SourceScanner プロジェクトを定義して実行するのではなく、1 つのプロジェクトに定義するべきです。

ソース管理システムおよびファイルのソースコードの統計の表示

SourceScanner を構成し、さまざまな Report Center プロジェクトに対してソース管理フィルターを定義すると、Report Center のレポート ビューとソース コード ビューにソース コード データに基づいて SCM の統計を表示することができます。詳細については以下のセクションを参照してください。

- 115 ページの「違反エクスプローラー」
- 132 ページの「カバレッジ エクスプローラー」
- 137 ページの「テスト エクスプローラー」
- 153 ページの「レポート」

ソースコードの参照 (ソース表示機能)

DTP を設定して Report Center に SCM データを含めることができます。DTP がリポジトリのソースを表示できるようにするには、Report Center と同じマシン上にリポジトリ クライアントをインストールする必要があります。次の表は、DTP と統合できる SCM の一覧です。

SCM	タスク
Accurev	Accurev クライアントがインストールされていること、および <code>accurev</code> コマンドがシステムパスに含まれていることを確認します。 問題のトラブルシューティングに関しては 355 ページの「ソース表示機能」を参照してください。 レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。
ClearCase	ClearCase クライアントがインストールされていること、および <code>cleartool</code> コマンドがシステムパスに含まれていることを確認します。 問題のトラブルシューティングに関しては 355 ページの「ソース表示機能」を参照してください。 レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。
CVS	CVS クライアントをインストールする必要はありません。 レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。

SCM	タスク
Jazz (Rational Team Concert)	<p>Jazz 2.0.0.2 がサポートされます。</p> <p>Jazz Plain Java Native Libraries が DTP マシンになければなりません。DTP は SourceScanner を介して Jazz Plain Java Native Libraries を発見することができます。</p> <p>Jazz コマンドライン クライアントは Jazz Eclipse Client に含まれるため、Jazz Eclipse Client が DTP マシンになければなりません。また、Jazz コマンドライン クライアントへのパスを PATH 環境変数に追加する必要があります。このパスは通常 [Jazz Directory]/scmtools/eclipse/ の下です。</p> <p>SourceScanner 5.4.2 が必要です。SourceScanner と DTP は同じマシン上になければなりません。</p> <p>レポート ビュー (153 ページの「レポート」) だけがサポートされます。</p>
GIT	<p>GIT クライアントがインストールされていることを確認します。</p> <p>git コマンドがシステム パスになければなりません。</p> <p>ローカル リポジトリ (Source Scanner で ROOT として設定) および DTP が同じマシン上になければなりません。</p> <p>レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。</p>
Microsoft Team Foundation Server	<p>違反エクスプローラーだけがサポートされます。115 ページの「違反エクスプローラー」を参照してください。</p>
Microsoft Visual SourceSafe	<p>違反エクスプローラーだけがサポートされます。115 ページの「違反エクスプローラー」を参照してください。</p>
Perforce SCM	<p>Perforce クライアントがインストールされていることを確認します。</p> <p>p4 コマンドがシステム パスになければなりません。</p> <p>レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。</p>
Serena Dimensions	<p>DM_ROOT 環境変数が設定されていて正しいことを確認します (たとえば <code>export DM_ROOT=/opt/serena/dimensions/12.2/cm</code>)。</p> <p>LD_LIBRARY_PATH 環境変数に Serena Dimensions libs へのパスが含まれていることを確認します (たとえば <code>export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/opt/serena/dimensions/12.2/cm/lib</code>)。</p> <p>レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。</p>

SCM	タスク
StarTeam	<p>Borland StarTeam SDK が [DTP_HOME]/lib/thirdparty ディレクトリにインストールされていることを確認します。SDK をインストールした後に DTP サービスを再開する必要があります。</p> <p>Borland StarTeam SDK は Borland の Web サイトから無料でダウンロードできます。</p> <p>レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。</p>
Subversion	<p>Subversion クライアントがインストールされていることを確認します。</p> <p>svn コマンドがシステム パスになければなりません。</p> <p>レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。</p> <p>問題のトラブルシューティングに関しては 355 ページの「ソース表示機能」を参照してください。</p>
Synergy CM	<p>Synergy クライアントがインストールされていること、ccm コマンドがシステム パスに含まれていることを確認します。</p> <p>レポート ビュー (153 ページの「レポート」) および違反エクスプローラー (115 ページの「違反エクスプローラー」) がサポートされます。</p> <p>問題のトラブルシューティングに関しては 355 ページの「ソース表示機能」を参照してください。</p>

PTC Integrity ソース管理コントロール拡張との統合

Parasoft DTP は PTC Integrity (MKS) 10.x と統合します。これは次の機能を可能にします。

- スコーピングと Parasoft Test タスク作成者の特定をサポートする
- Parasoft コード レビューの実行をユーザーに許可する
- Parasoft Test タスクからソース コードを参照する
- コード レビューの自動化プロセスを容易にする

PTC Integrity 10.4 を使って以下の統合サポートがテストされ、確認されています。ただし、この統合は PTC Integrity 10.0 ~ 10.4 をサポートする MKS API 4.10 をサポートするために実装されました。

統合用パッケージ

Parasoft は、MKS 統合のために次のパッケージを提供しています。

- `xtestMKS.jar`: PTC Integrity 10.x をサポートするための Parasoft ソース管理の実装
- `mksapi.jar`: PTC Integrity Java API

要件

- PTC Integrity 10.x クライアントをインストールする必要があります。Parasoft は、情報を取得するために、PTC Integrity クライアントの Source Integrity (SI) コマンドラインと Java API の両方を使用します。
- ツールが使用するプロジェクト用のサンドボックスを初めに作成する必要があります。すべてのファイルをこのサンドボックスに置く必要があります。
- Parasoft Test 9.5.10 (Jtest、C++test、SOAtest) 以上を必ずインストールする必要があります。
- Linux にインストールされた Parasoft DTP。Windows サーバー上の DTP については、この統合は現在サポートされていません。

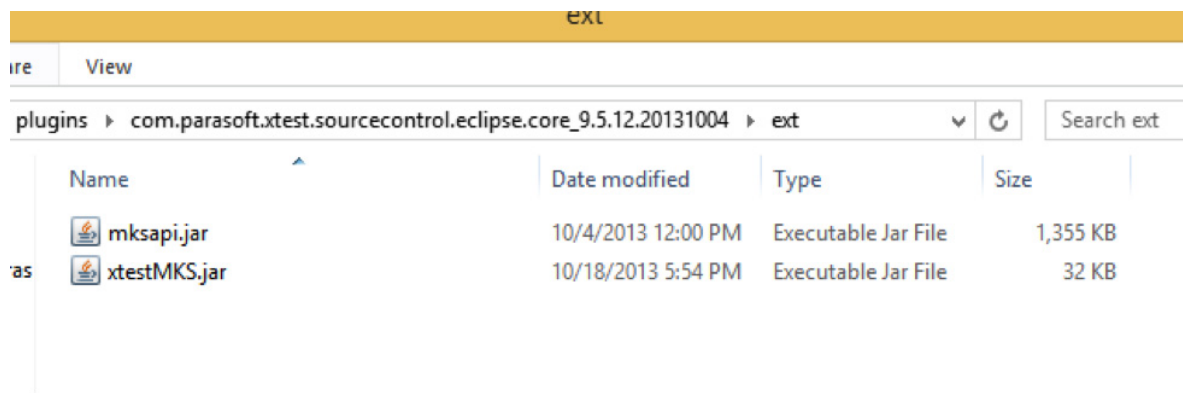
インストールと設定

パッケージは Parasoft Test と DTP でインストールされ、設定されます。

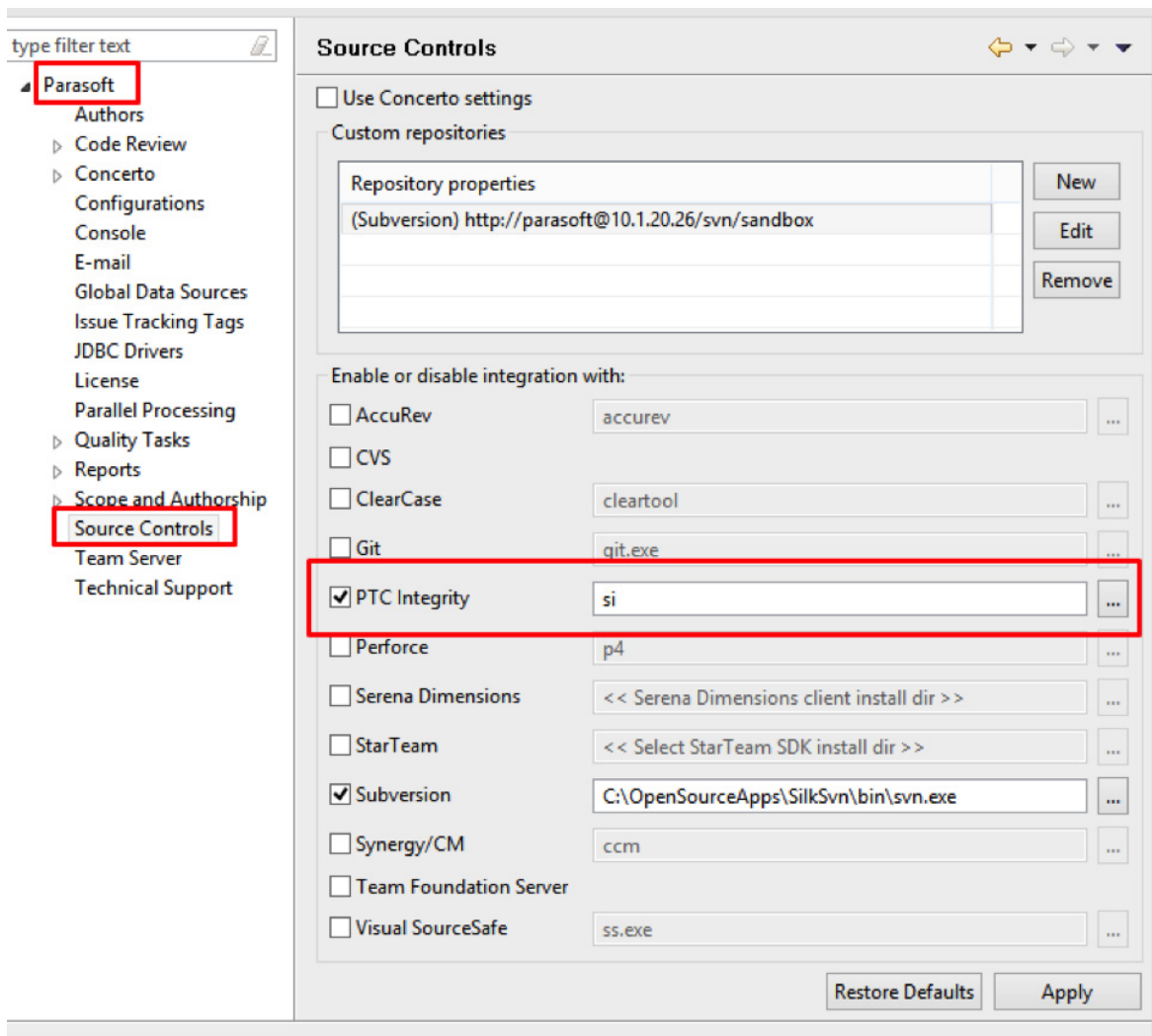
Parasoft Test へのインストール

1. インストール ディレクトリを開きます。

- Windows 32-bit スタンドアロン : C:\Program Files\Parasoft\Test\9.5
 - Windows 64-bit スタンドアロン : C:\Program Files (x86)\Parasoft\Test\9.5
 - Windows 32-bit プラグイン : C:\Program Files\Parasoft\Test for Eclipse\9.5
 - Windows 64-bit プラグイン : C:\Program Files (x86)\Parasoft\Test for Eclipse\9.5
 - Linux スタンドアロン : \$HOME/parasoft/test/9.5
 - Linux プラグイン : \$HOME/parasoft/test for eclipse/9.5
2. [INSTALL_DIR]/plugins/
com.parasoft.xtest.sourcecontrol.eclipse.core_<version> ディレクトリに
ext というディレクトリを作成します。
 3. ext ディレクトリに統合パッケージをコピーします。



4. ツールを起動し、[Parasoft] > [設定] > [ソース管理] を選択して PTC Integrity オプションを利用できることを確認します。



Parasoft DTP へのインストール

1. MKSjars.zip ファイルの内容を、DTP がインストールされているシステム中の一時的なディレクトリに展開します。
2. \$DTP_HOME/plugins ディレクトリに scontrol というディレクトリを作成します。
3. 任意のエディターで \$DTP_HOME/bin/reportserver.sh スクリプトを開きます。
4. CATALINA_OPTS という行を探し、次の行を追加します。


```
-Dscontrol.ext.dir=$PST_HOME/plugins/scontrol \
```
5. スクリプト ファイルを保存し、エディターを終了します。

6. \$PST_HOME/bin/concertoconsole.sh スクリプトを使って DTP を再起動します。

```
# Needs to be exported to be seen by catalina.sh
export JAVA_HOME
export JAVA_OPTS="-D$PCC_RECOGNITION"

export CATALINA_OPTS="-Djava.ext.dirs=$PST_HOME/lib/rs:$PST_HOME/lib/ext \
    $JAVA_MEM \
    -Dpcc.dir=$PST_HOME -Dinstall.dir=$GRS_HOME -Dtcn.dir=$TCM
_HOME -Dls.dir=$LS_HOME \
    -Dcom.parasoft.xtest.logging.config.file=$PST_HOME/bin/log
_config.xml \
    -Dscontrol.ext.dir=$PST_HOME/plugins/scontrol \
    -Dsource.scanner.log=source_scanner \
    -Djava.awt.headless=true"

export LICENSE_SERVER_HOME=$LS_HOME

if [ "$1" = "start" ] ; then

    checkPortAvailable

    echo "*** Starting catalina at" `date` >> $RS_OUT_LOG
    exec $CATALINA_HOME/bin/catalina.sh run >> $RS_OUT_LOG 2>> $RS_ERROR_LOG
"reportserver.sh" 148L, 3707C                                128,23                92%
```

Parasoft Test での設定とセットアップ

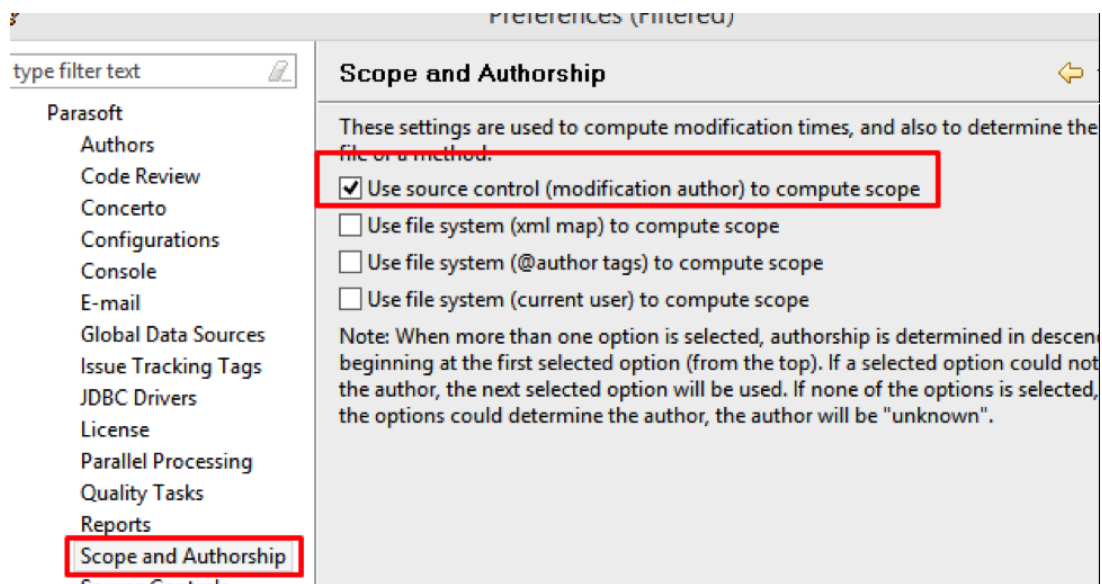
Linux/Solaris の場合、設定の残り部分に進む前に以下のステップを実行します。

- \$INSTALLDIR/bin ディレクトリを PATH の先頭に必ず追加します。
- \$INSTALLDIR/lib/linux (Solaris の場合、\$INSTALLDIR/lib/solaris) を指すように LD_LIBRARY_PATH を必ず設定します。

\$INSTALLDIR/lib/<linux/solaris> ディレクトリには 32-bit と 64bit の両方のライブラリがあります。JVM に基づいて MKS API が適切なライブラリをロードします。

1. Parasoft Test アプリケーションを起動します。
2. [Parasoft] > [設定] > [スコープと作成者] を選択します。
3. [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] チェックボックスをオンにします。

4. 他のすべてのチェックボックスをオフにして [適用] クリックします。



5. [Parasoft] > [設定] > [Parasoft] > [ソース管理] を選択し、PTC Integrity オプションが有効であること、および si コマンドが PATH に設定されていることを確認します。si が PATH がない場合、必ず絶対パスで si を PATH に追加してください。
6. [カスタム リポジトリ] セクションで [新規] をクリックし、[プロパティ] ウィンドウに次のオプションを入力します。

serverport: PTC Integrity ポート (デフォルトは 7001)

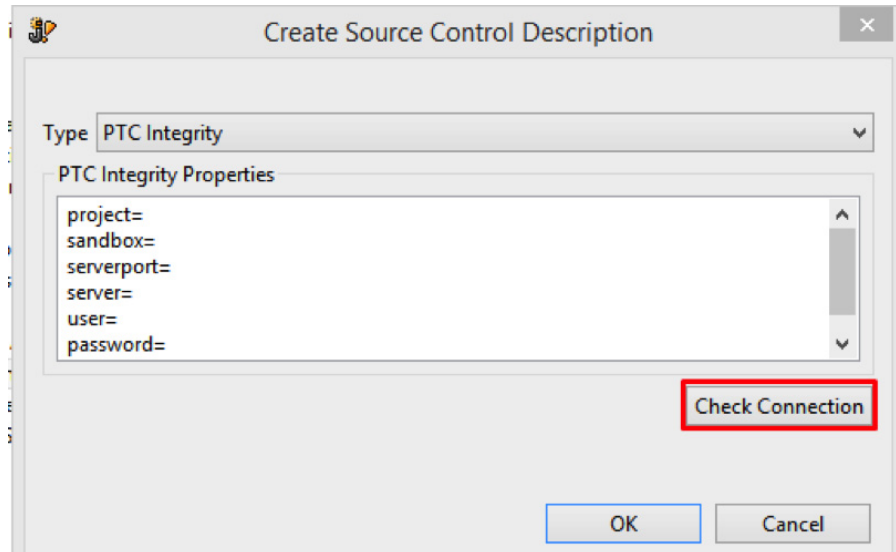
server: PTC Integrity サーバー名

user: PTC Integrity に対して認証されるユーザー名

password: PTC Integrity に対して認証されるパスワード

sandbox: ローカル リソースが同期化されるサンドボックスの場所 (パス中で project.pj で終わる)

project: サーバー上の PTC Integrity プロジェクトの場所 (例: /Project XZY/project.pj)

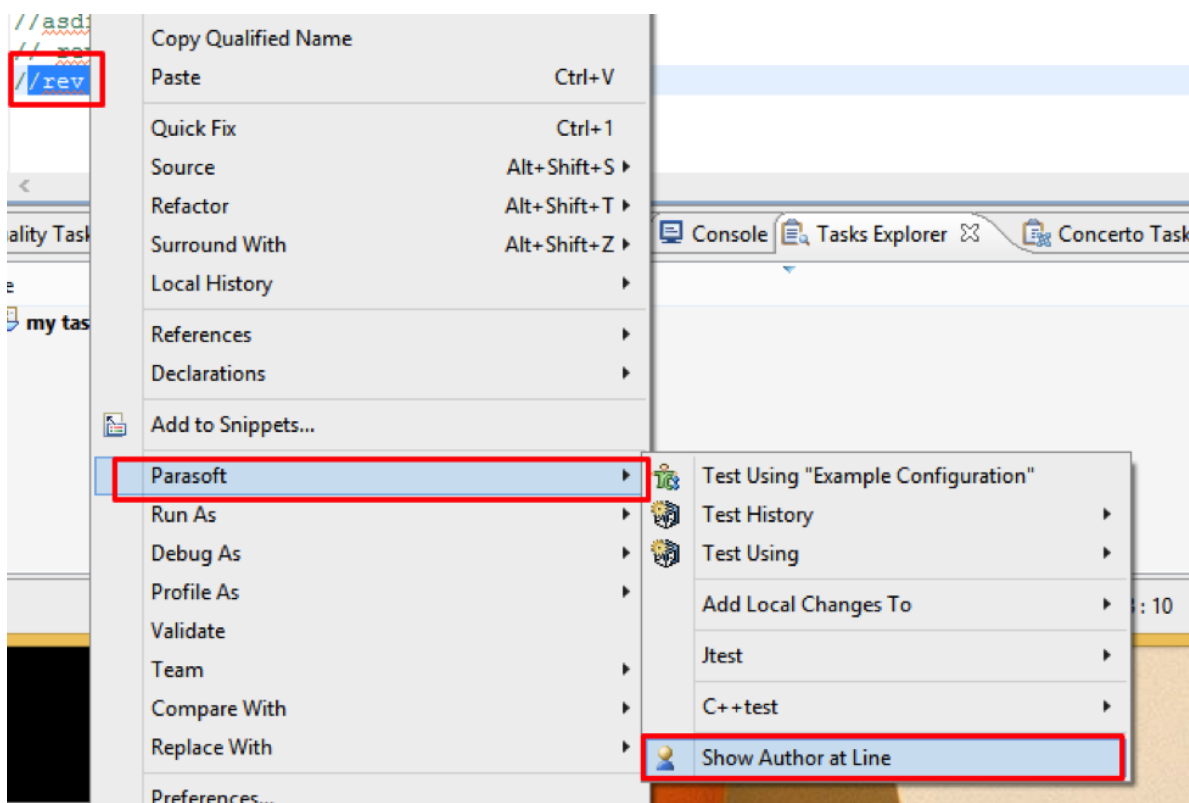


7. [接続のチェック] をクリックし、オプション設定を確認します。エラーがある場合、PTC Integrity のエラーメッセージが返され、ダイアログに表示されます。
8. [OK] をクリックして、設定を保存して終了します。

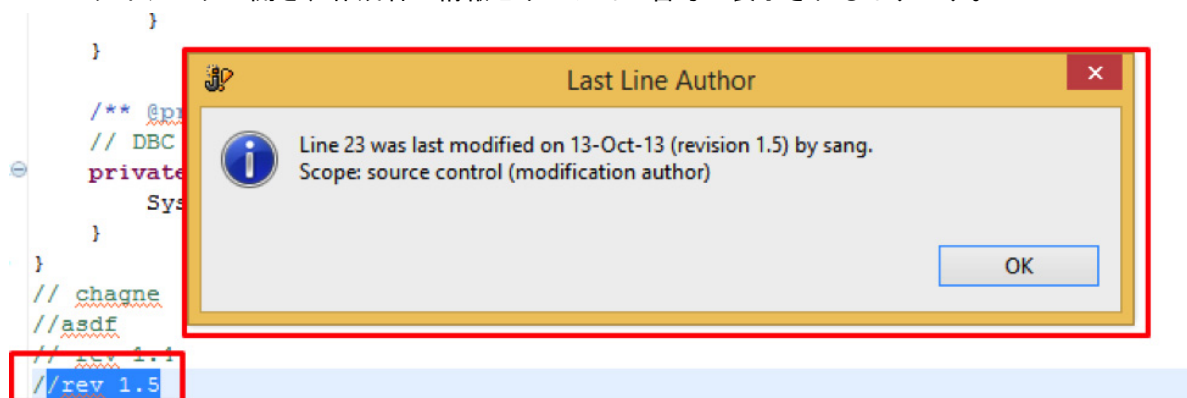
Parasoft Test での設定の確認

1. サンドボックスのプロジェクトを Parasoft Test ソリューションにインポートします。
2. ファイルを開き、行を右クリックします。

3. [Parasoft] メニューの [Show Author at Line] をクリックします。



4. ダイアログが開き、作成者の情報とリビジョン番号が表示されるはずですが。



DTP と SourceScanner での PTC Integrity の設定

設定に進む前に、SourceScanner と DTP を実行するマシンに PTC Integrity (MKS) クライアントがインストールされていて正しく動作していることを確認してください。

Linux/Solaris の場合、設定の残り部分に進む前に以下のステップを実行します。

- \$INSTALLDIR/bin ディレクトリを PATH の先頭に必ず追加します。
- \$INSTALLDIR/lib/linux (Solaris の場合、\$INSTALLDIR/lib/solaris) を指すように LD_LIBRARY_PATH を必ず設定します。

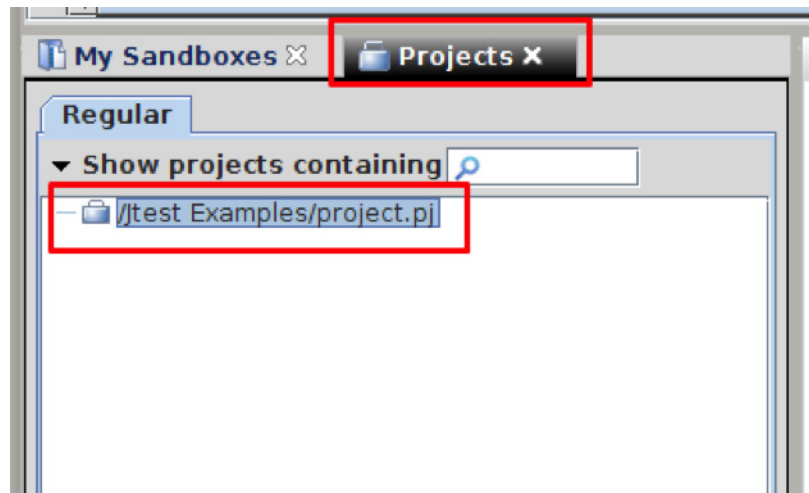
`$INSTALLDIR/lib/<linux/solaris>` ディレクトリには 32-bit と 64bit の両方のライブラリがあります。JVM に基づいて MKS API が適切なライブラリをロードします。

Windows の場合、SYSTEM 環境の PATH に必ず `%INSTALLDIR%/bin` を追加してください。

Bash の場合 :

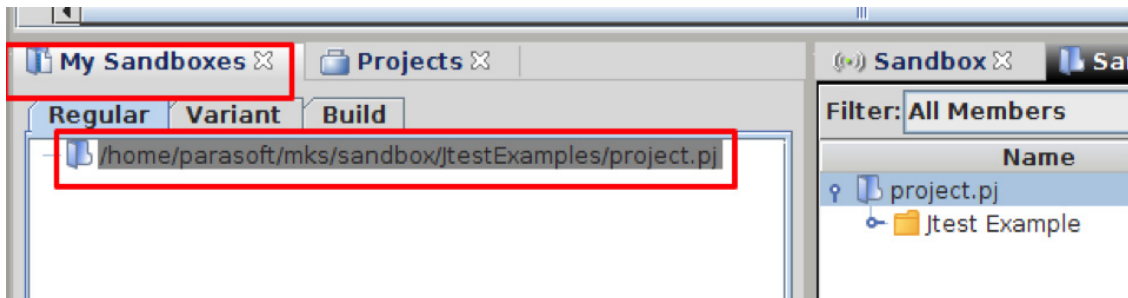
```
PATH=/home/Integrity/IntegrityClient10/bin:$PATH
LD_LIBRARY_PATH=/home/Integrity/IntegrityClient10/lib/
linux:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH PATH
```

1. `si gui` コマンドを開始して Integrity Client GUI を開きます。
2. [Projects] タブの Identify プロジェクトパスが次のように表示されます。



この値は、SourceScanner で Project Root として使用されます。

3. このシステム用のサンドボックスを作成し、サンドボックスの値を記録します。



この値は、SourceScanner でサンドボックスの場所として使用されます。

4. DTP のインストール ガイドに従って、DTP をインストールします。Linux で DTP を起動する前に、PATH と LD_LIBRARY_PATH が SHELL にロードされることを確認してください。
5. `SDLCExtensions-5.7.3.zip` ファイルを展開して SourceScanner をインストールします。

Windows の場合 :

1. zip ファイルを `C:\Parasoft` に展開します。`C:\Parasoft\SDLCExtensions` ディレクトリが作成されます。

2. PROSERVE_HOME 環境変数を C:\Parasoft\SDLCExtensions に設定します。

Linux/Solaris の場合 :

1. \$HOME/parasoft ディレクトリに zip ファイルを展開します (\$HOME/parasoft/SDLCExtensions ディレクトリが作成されます)。
2. bin ディレクトリに移動し、すべてのスクリプト実行可能ファイルに対して `chmod a+x *.sh` を実行します。
3. \$HOME/.proserve.rc を編集し、次の行を追加しました。


```
export JAVA_HOME=<JAVA_RUNTIME_INSTALL_DIR>
export PROSERVE_HOME=<where SDLCExtensions are installed>
($HOME/parasoft/SDLCExtensions)
```
4. DTP または SourceScanner を起動する前に、\$HOME/.bashrc で PATH および LD_LIBRARY_PATH が保存されてロードされることを確認します。
6. SSGUI.sh (Linux/Solaris) または SSGUI.cmd (Windows) を開始し、SourceScanner GUI を起動します。
7. SDLCExtensions のドキュメントに従って、SourceScanner をセットアップします。以下のセクションでは、PTC Integrity に関連する情報だけを説明します。

PTC (MKS) Integrity プロジェクトを作成する

1. プロジェクトの下で、ソース管理の種類として MKS を選択します。
2. プロジェクト名を入力します。これはリポジトリ ID として DTP で使用されます。
3. ルートは、前述のステップ 2 で得られたサーバーのプロジェクト パスになります。
4. サンドボックスの場所は、前述のステップ 3 の値になります。
5. その他の設定と情報については、『SDLCExtensions ユーザーズ ガイド』を参照してください (注意: SDLCExtensions ドキュメントの MKS の参考情報は古い内容であり、6.0 リリースで更新される予定です)。
6. SourceScanner 実行してデータを DTP にポピュレートします。

サードパーティのツールから DTP へのテスト結果の送信

DTP には、カスタムおよびサードパーティのテスト結果を DTP に送信するために使用できるオープン API が用意されています。テスト結果を DTP に送って、たとえば Report Center ダッシュボードで可視化するなど、開発プロセスでテスト結果を利用することができます。

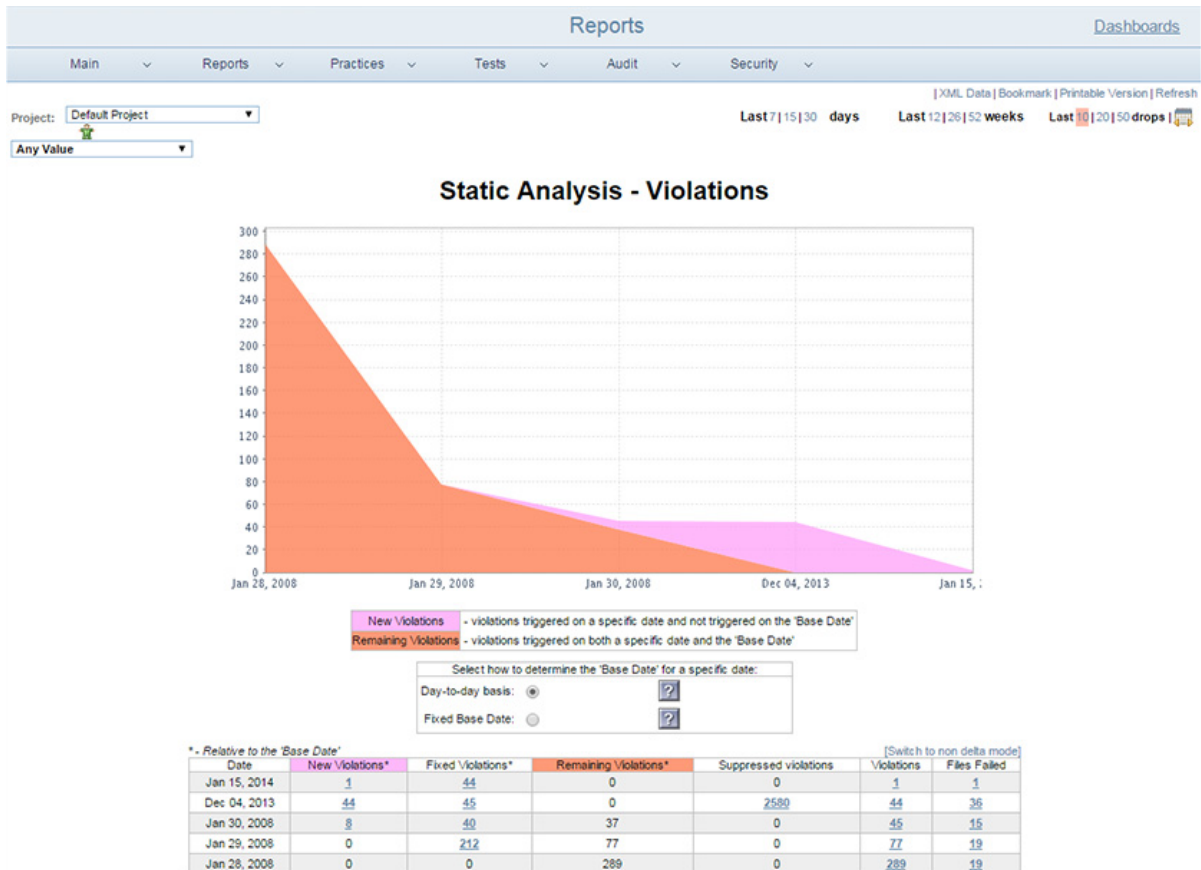
この API はオープンソースのバンドルとして DTP に付属しています。詳細については次の URL を参照してください。

<http://sourceforge.net/projects/psfconcertoapi/?source=directory>

Microsoft Excel へのレポートのインポート

[XML データ] オプションを使用すると、Report Center のレポートに表示されているデータをエクスポートし、Microsoft Excel などの表計算アプリケーションにインポートすることができます。独自のテーブル、レポート、グラフを作成することができます。

1. レポートを開き、[XML データ] リンクをクリックします。



一部の最上位レベルのレポートには [XML Data] リンクがあります。その場合、データをエクスポートする特定のグラフまでドリルダウンする必要があります。

2. ブラウザーの設定によっては、ファイルを開くか保存するかを選択を求められます。

Do you want to open or save **xreport.xml** (1.48 KB) from **dtp2.parasoft.com**?

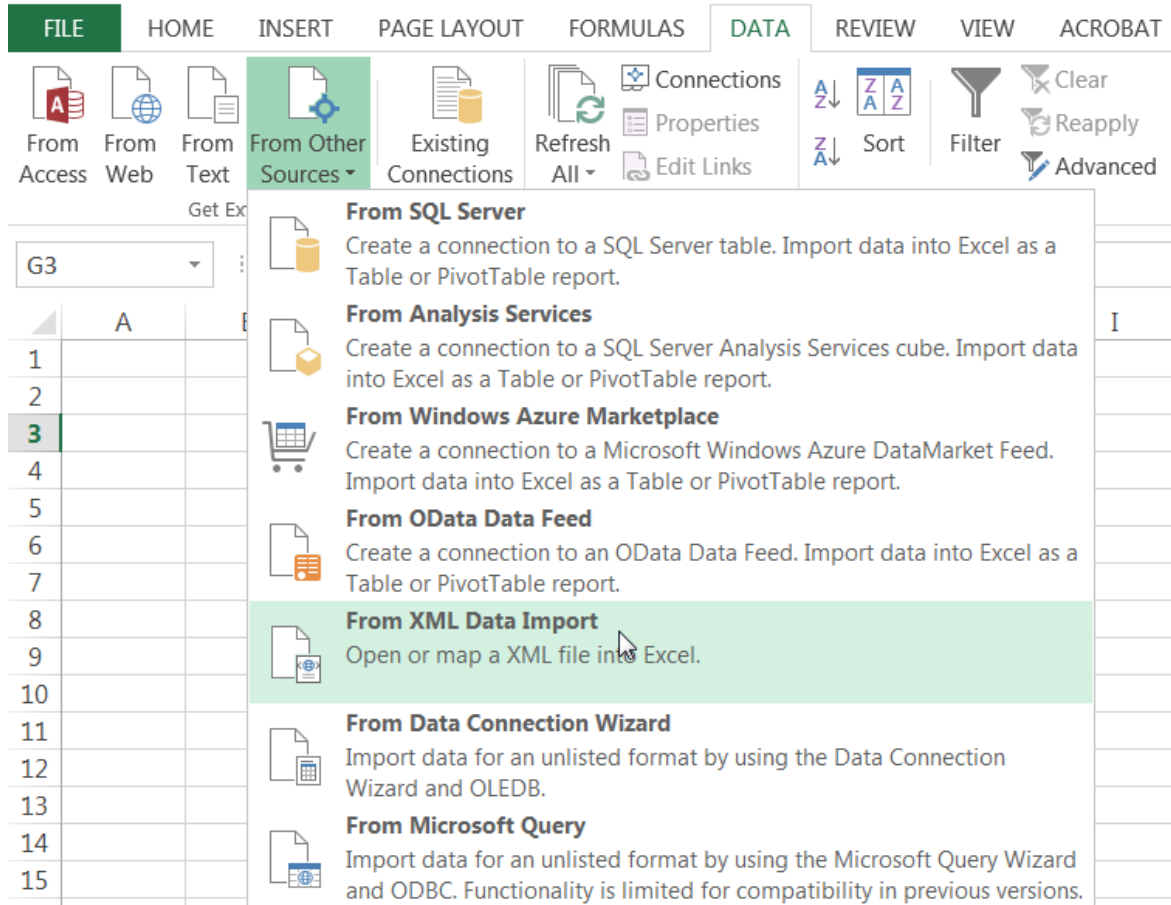
Open

Save

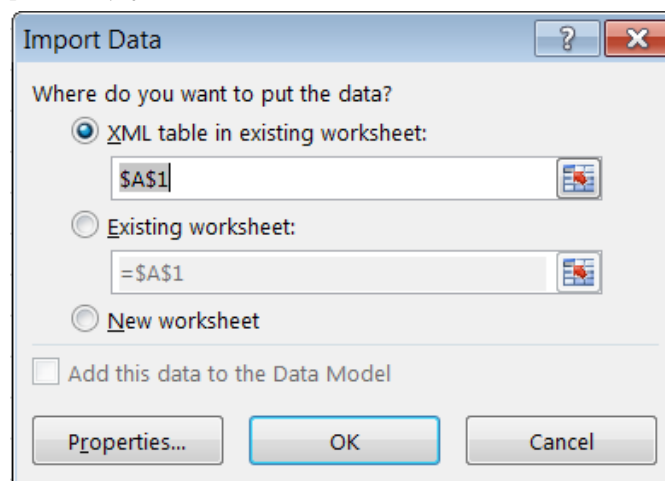
Cancel

3. 表計算アプリケーションを起動し、ダウンロードした XML ファイルをインポートします。この例では Microsoft Excel を使用します。
4. [ファイル] > [新規作成] > [空白のブック] を選択します。

5. [データ] タブをクリックし、[外部データの取り込み] リボンで [その他のデータ ソース] > [XML データ インポート] を選択します。



6. スプレッドシート中でデータを配置する場所を指定して [OK] をクリックし、レポートのインポートを完了します。



Java API

Development Testing Platform の Java API を利用すると、Development Testing Platform の機能を拡張することができます。Development Testing Platform には、Bugzilla および JIRA バグ トラッキング システムと統合する機能が組み込まれていますが、他のバグ トラッキング システムを使用している組織は、Java API のバグ スキャナーを拡張して、独自のカスタム バグ スキャナーを作成できます。また、Java API の要求スキャナーを拡張して、独自のカスタム要求スキャナーを作成することもできます。

Java API のドキュメントを参照するには、ブラウザのアドレス バーに `https://host:port/grs/java-api` と入力します。通常、Development Testing Platform がポート番号 80 にデプロイされている場合は、URL は `https://host/grs/api` です。80 番ポートが使用されていない場合、通常は 8443 にデプロイされています。Java API の使用に関する詳細は、270 ページの「手動による BTS/RMS Scanner の構成」および 274 ページの「カスタム RMS Scanner」を参照してください。

DTP カスタム プロセッサ用の Java API は <http://build.parasoft.com/maven> にあります。利用可能なビルド成果物には、コンパイル済みの jar ファイルおよびパッケージされた javaodc の jar ファイルが含まれています。以下のプロパティは、これらのファイルに共通です。

- groupId: com.parasoft.dtp
- artifactId: com.parasoft.api.dtp.processors
- version: 5.1.0
- groupId: com.parasoft.dtp
- artifactId: com.parasoft.api.dtp.processors
- version: 5.1.0

Report Center のインストール

このセクションの内容：

- Report Center の管理 (Windows)
- Report Center の管理 (Linux)

Report Center の管理 (Windows)

古い Development Testing Platform (DTP) または Report Center がインストールされている場合、MySQL または Report Center のアンインストール、古い Report Center データベースの消去、または Report Center のアップグレードを行うことを推奨します。

DTP または MySQL のアンインストール

DTP をアンインストールせずに Report Center だけをアンインストールすることはできません。また、MySQL を削除する前に DTP/Report Center を削除する必要があります。

DTP/Report Center をアンインストールするには、次の操作を行います。

1. Windows の [スタート] メニューをクリックして [コントロール パネル] をクリックします。
2. [プログラムの追加と削除] をダブルクリックします。[プログラムの追加と削除] ウィンドウが表示されます。
3. Report Center を選択して [削除] をクリックします。画面の操作指示に従います。

MySQL をアンインストールするには、次の操作を行います。

1. Windows の [スタート] メニューをクリックして [コントロール パネル] をクリックします。
2. [プログラムの追加と削除] をダブルクリックします。[プログラムの追加と削除] ウィンドウが表示されます。
3. MySQL を選択して [削除] をクリックします。画面の操作指示に従います。

Report Center/Project Center データベースの消去

MySQL をアンインストールせずにデータベースを新しくしたい場合、Report Center データベースを消去してから新しいデータベース構造を作成することができます。

注意： Report Center データベースを消去する前に MySQL サービスが実行中である必要があります。

データベースを消去するには、次の操作を行います。

1. Data Collector を停止します。
2. DTP Server を停止します。
3. mysql ユーティリティを実行します。
4. プロンプトで次のコマンドを実行します。

```
drop database GRS
```

5. システムを再起動します。
6. 『Development Testing Platform クイック インストール ガイド』の説明に従って新しいデータベースを作成します。

Report Center の管理 (Linux)

Report Center を適切にインストールしたら、`dtppconsole.sh` を実行してメインメニューを表示することができます。`dtppconsole.sh` は `DTP_HOME/bin` ディレクトリにあります。標準の DTP ディストリビューションをインストールした場合、次のオプションを利用できます。

```
Development Testing Platform - Configuration Manager
Copyright (C) by Parasoft Corporation.

MAIN MENU
=====

Options:
(1) Development Testing Platform Server (Report Center, Project Center,
Team Server, License Server)
(2) Data Collector
(3) Status
(4) Database configuration
(5) System administration
(6) Show Machine Id
(q) Exit to system
```

- (1) を選択すると、Development Testing Platform Server のメニューが表示されます。340 ページの「Development Testing Platform Server メニュー」を参照してください。
- (2) を選択すると、Data Collector のメニューが表示されます。341 ページの「Data Collector メニュー」を参照してください。
- (3) を選択すると、エラーのチェック、Parasoft サービスが実行中かどうかの確認、および使用ディスク容量の確認を行うことができます。
- (4) を選択すると、データベース構成のメニューが表示されます。
- (5) を選択すると、システム構成のメニューが表示されます。
- (6) を選択すると、マシン ID が表示されます。
- (q) を選択すると、メインメニューが終了します。

Development Testing Platform Server メニュー

Development Testing Platform Server のさまざまなパラメーターを設定することができます。

警告！ 各オプションの値をデフォルトのままにすることを推奨します。

オプション	結果
(1) [Run /Stop] Service	DTP サービスを開始または停止します。
(q) Exit to previous menu	プロンプトで「q」を入力すると、DTP Server のメニューが終了し、メインメニューが表示されます。

Data Collector メニュー

オプション	結果
(1) [Run /Stop] Service	Data Collector サービスを開始または停止します。
(q) Exit to previous menu	プロンプトで「q」を入力すると、DTP Server メニューが終了し、メインメニューが表示されます。

System Administration メニュー

Option	Result
(1) Enable/Disable administration mailing	ディスク容量が少なくなったときに Report Center 管理者に警告する機能の有効 / 無効を設定できます。この機能を利用するには、ローカルの SMTP サーバーが実行中であり、適切に構成されていなければなりません。デフォルトは "無効" です。
(2) Change administrator address	ディスク容量の低下についての通知を受け取る担当者の電子メール アドレスを指定できます。
(3) Change risky disk space usage [90%]	ディスク使用量を設定できます。ディスク容量がこの値に達すると、Report Center の管理者に電子メールでディスク容量の低下が通知されます。
(q) Exit to previous menu	DTP Server メニューが終了し、メインメニューが表示されます。

MySQL の root ユーザー パスワードの変更

デフォルトでは、mysql データベースの root ユーザーのパスワードは空です。つまり、パスワードの入力を求められたときに、Enter キーを押すだけで root ユーザーとして mysql データベースへ

のアクセスが許可されます。たとえば dtpconsole.sh の Database Menu は、このパスワードがない状況を利用して、初期データベースを作成するために mysql に接続しようとします。しかし、mysql の root ユーザーにパスワードを設定するのは良いプラクティスであり、ネットワークを介して接続する場合は特にそうです。パスワードを設定する場合、接続時に使用するパスワードを dtpconsole.sh が把握できなければなりません。そうでなければ、空のパスワードが使用されて接続に失敗します。ユーザーは Database Menu を使って適切なパスワードを入力することができます。

1. Main Menu のプロンプトで 「3」 を入力して Database メニューを表示します。

Database メニューで利用できるオプションは次の表のとおりです。

オプション	結果
(1) Create initial database	プロンプトで 「1」 を入力すると、初期データベースが作成されます。
(2) Set MySQL root password	プロンプトで 「2」 を入力すると、mysql root パスワードを入力できます。このパスワードは mysql データベースに接続するときに dtpconsole.sh によって使用されます (デフォルトはパスワードなしです)。
(q) Exit to previous menu	プロンプトで 「q」 を入力すると、Database Menu が終了し、Main Menu が表示されます。

2. Database Menu から [Set mysql root password] オプションを選択します。

使用できる Development Testing Platform 製品の確認

1. Parasoft Development Testing Platform のロゴをクリックします。
2. [License configuration] リンクをクリックします。



有効化されていてライセンスが許可されている Development Testing Platform 製品の一覧が表示されます。

License status: [License OK.](#)

Development Testing Platform version: DTP 5.0
Machine Id: LINUX2-

License details:

Expiration date: 05 Dec 2014

Password:

Features:

- DTP
- DTP Report Center
- DTP Project Center
- DTP Policy Center
- DTP Team Server
- DTP License Server
- DTP License Server unlimited in subnet

Max tokens: 1

[\[Configure license\]](#)

データの移行

Development Testing Platform (バージョン 5.x) または Concerto (バージョン 4.x) の古いバージョンからデータをバックアップし、新しいバージョンに移行してください。以下のタスクを実行してデータを移行することができます。

- License Server からのデータの移行。次のいずれかを行います。
 - License Server 1.0 または 2.0 からの移行, or
 - License Server 2.x から新しいバージョンへのデータ移行
- Team Server 構成の移行
- Team Server データの移行

License Server 1.0 または 2.0 からの移行

License Server のライセンス データは /LicenseServer/.psrc ディレクトリに格納されています。License Server 1.0 または 2.0 から Concerto 1.x にデータをインポートするには、次の操作を行います。

1. .psrc ファイルをバックアップします。
2. 以前のバージョンの License Server および Concerto をアンインストールします。
3. Concerto 1.x をインストールします。
4. /LicenseServer/conf/.psrc.xml が存在することを確認してから、License Server が起動時にこのファイルを使用しないよう、ファイル名をたとえば .psrc.xml.bak などに変更します。
5. License Server が以前のライセンス構成を使用するよう、ステップ 1 でバックアップした .psrc ファイルを /LicenseServer/conf/ ディレクトリにコピーします。
6. Concerto を実行します。

起動時に License Server は .psrc ファイルからライセンス構成をインポートし、そのデータに基づいて新しい .psrc.xml ファイルを作成します。

License Server 2.x から新しいバージョンへのデータ移行

License Server 2.x から新しいバージョンの Development Testing Platform にデータをインポートするには、次の操作を行います。

1. DTP_HOME/LicenseServer/conf/ ファイルをバックアップします。
2. 以前のバージョンをアンインストールします。
3. Development Testing Platform をインストールします。

詳細については『Development Testing Platform クイック インストール ガイド』を参照してください。

4. バックアップしたファイルで `DTP_HOME/LicenseServer/conf` を上書きします。

Repair オプション (推奨) でアップグレードする場合は、バックアップを使用する必要はありません。なぜなら、以前のデータと構成は、新しいバージョンで使用するために自動的に保持されるからです。ただし、新しいバージョンのインストールが失敗し、以前のバージョンを再インストールしてデータを復元しなければならない場合に備えて、バックアップを保管することを推奨します。

Team Server 構成の移行

以前のバージョンの Team Server 構成を復元するには、次の操作を行います。

1. `DTP_HOME/tcm/conf/` ファイルをバックアップします。
2. 以前のバージョンの Team Server/Concerto をアンインストールします。
3. Concerto 1.x をインストールします。

詳細については『Development Testing Platform クイック インストール ガイド』を参照してください。

4. バックアップしたファイルで `DTP_HOME/tcm/conf` を上書きします。

Repair オプション (推奨) でアップグレードする場合は、バックアップを使用する必要はありません。なぜなら、以前のデータと構成は、新しいバージョンで使用するために自動的に保持されるからです。ただし、新しいバージョンのインストールが失敗し、以前のバージョンを再インストールしてデータを復元しなければならない場合に備えて、バックアップを保管することを推奨します。

Team Server データの移行

以前の Team Server データを復元するには、次の操作を行います。

1. `DTP_HOME/tcm/storage/` ファイルをバックアップします。
2. 以前のバージョンの Team Server/Concerto をアンインストールします。
3. Concerto 1.x をインストールします。

詳細については『Development Testing Platform クイック インストール ガイド』を参照してください。

4. Copy and paste the backup to the new `DTP_HOME/tcm/storage` location.

Repair オプション (推奨) でアップグレードする場合は、バックアップを使用する必要はありません。なぜなら、以前のデータと構成は、新しいバージョンで使用するために自動的に保持されるからです。ただし、新しいバージョンのインストールが失敗し、以前のバージョンを再インストールしてデータを復元しなければならない場合に備えて、バックアップを保管することを推奨します。

MySQL サーバーのアップグレード

既存の Report Center データベースがあって、このデータベースを残したい場合に MySQL サーバーをアップグレードすることができます。アップグレードの前に、必ず Report Center データベース ファイルをバックアップしてください。Report Center データベースのバックアップの詳細については『Development Testing Platform クイック インストール ガイド』を参照してください。MySQL をアップグレードする方法については、MySQL ドキュメント ライブラリを参照してください。

<http://dev.mysql.com/doc/>

Configuration Manager の内部詳細 (Linux/Solaris)

このセクションでは、Configuration Manager コマンドライン プログラムを使用するときに実行される、内部的な DTP スクリプト (DTP_HOME/bin/dtpconsole.sh) について説明します。

このセクションの内容：

- Web サーバーの実行
- Data Collector の実行

Web サーバーの実行

DTP Web サーバーを実行するために (1) を選択したとき、dtpconsole.sh は次のスクリプトを実行します。

- DTP_HOME/bin/reportserver.sh

構成ファイル \$DTP_HOME/.server_params も次の行によって更新されます。

- service=ENABLED

エラー メッセージを含め、情報を持つ DTP 製品のすべてのログは、DTP_HOME\logs\ ディレクトリに保存されます。メイン ログ ファイルは rs.log です。DTP Server のプロセス ID は DTP_HOME/grs/log/rs.pid に格納されます。

他のログファイルの詳細については 641 ページの「検証ステップ 3: ログ ファイル中に未知の例外またはメッセージがない」を参照してください。

構成ファイルで service=ENABLED が定義されている限り、DTP Server プロセスの最中に問題が発生した場合、DTP のインストール時にインストールされた cron ジョブがプロセスをモニタリングし、DTP Server を再起動します。

Data Collector の実行

Data Collector を実行するために (1) を選択したとき、dtpconsole.sh は次のスクリプトを実行します。

- DTP_HOME/bin/datacollector.sh.

構成ファイル \$DTP_HOME/.collector_params も次の行によって更新されます。

- service=ENABLED

datacollector.sh スクリプトは Data Collector を実行します。エラー メッセージを含め、Data Collector のすべてのログは DTP_HOME\logs\dc* ファイルに保存されます。メインのログ ファイルは dc.log です。

Data Collector のプロセス ID は DTP_HOME/grs/log/dc.pid に格納されます。構成ファイルで service=ENABLED が定義されている限り、データ収集プロセスの最中に問題が発生した場合、DTP

のインストール時にインストールされた cron ジョブがプロセスをモニタリングし、Data Collector を再起動します。

Data Collector が実行され、Parasoft 製品やサードパーティ製ツールから収集されたデータがデータベースに保存されます。

Linux での初期データベースの作成とデータベースのアップグレード (コマンド ライン メニューからの実行)

Linux で初期データベースを作成するには、次の操作を行います。

1. 次のコマンドを入力します。

```
dtppconsole.sh
```

スクリプトがパスに含まれていなければなりません。スクリプトは \$DTP_HOME/bin ディレクトリにあります。

コマンドを実行すると、次のように Development Testing Platform Configuration Manager Main Menu が表示されます。

```
MAIN MENU
```

```
=====
```

```
Options:
```

```
    (1) Development Testing Platform Server (Report Center, Project  
        Center, Team Server, License Server)
```

```
    (2) Data Collector
```

```
    (3) Database (MySQL only)
```

```
    (4) Status
```

```
    (5) System Administration
```

```
    (q) Exit to system
```

```
Choose one:
```

2. プロンプトに「3」を入力して [Database] オプションを選択します。Main Menu の他のオプションの詳細については 340 ページの「Report Center の管理 (Linux)」を参照してください。

「3」を入力すると、Database Menu が表示されます。

```
DATABASE MENU
```

```
=====
```

```
Options:
```

```
    (1) Create initial database
```

```
    (2) Set mysql root password
```

```
    (q) Exit to previous menu
```

```
Choose one:
```


3. プロンプトに「1」を入力して [Create initial database] オプションを選択します。
Database Menu の他のオプションの詳細については 341 ページの「MySQL の root ユーザー パスワードの変更」を参照してください。

注意：デフォルトでは、MySQL データベースは空の root パスワードでインストールされません。空の root パスワードではデータベース サーバーに接続できない場合、管理スクリプトは正しい MySQL root パスワードを入力するようプロンプトを表示します。

4. 新規インストールの場合、データベースをローカル マシンに作成するかどうかを訊ねられます。[Yes] を選択します。

アップグレードの場合：

アップグレードを実行するかどうかを訊ねられます。[Yes] を選択します。

プロンプトが表示され、root のパスワードを入力するよう求められます。

5. MySQL の構成で使用するパスワードを入力します。

スクリプトは root アカウントを使用して MySQL サーバーに接続し、初期データベースを作成するスクリプトを実行します。

注意：問題が発生した場合、以下のどちらかの可能性があります。

- 1) MySQL の構成に不足がある。
- 2) データベースの権限の問題がある。

6. (オプション) 念のためにレポートがすべて最新の状態かどうかを検証するよう要求できます。この操作は Report Center の Web インターフェイスで行います。変更の必要があるケースは少ないでしょう。

アップグレードの場合：

念のために次のコマンドを実行してもう一度データベースの整合性を検証すると良いでしょう。

```
mysqlcheck -u grs -pgrs GRS
```

7. (オプション) Report Center ディレクトリの容量を増やすため、またはその他の理由で MySQL 内にデータ ディレクトリを移動したい場合、ここで移動を行うと良いでしょう。

注意：MySQL へのアクセスで問題が発生した場合、次の SQL スクリプトを使用して初期データベースを作成できます。

```
DTP_HOME/grs/db/create_db_mysql.sql
```

この操作を行う時点で Report Center は正常にインストール済みでなければなりません。Report Center は DTP のインストール プロセスでインストールされます。

crontab による自動起動の設定 (Linux)

通常、Linux で Development Testing Platform (DTP) をインストールする場合には crontab が変更されます。DTP Server を自動的に起動する `cronguard.sh` をシステム リブートのたびに呼び出すよう、crontab が構成されます。

サーバーの起動時における DTP の自動起動は crontab で実装されます。なぜなら、crontab ではルート アクセスが必要ないからです。UNIX の `/etc/init.d` サービスで実装したなら、ルート アクセスが必要になります。

DTP のインストール時に crontab の構成をスキップした場合、以下の操作を行って手動で crontab を設定してください。

1. DTP をインストールしたユーザーとしてログインします。
2. `crontab -e` コマンドを入力します。
3. エディターに以下を入力します。`$DTP_HOME` には、DTP がインストールされているディレクトリをフルパスで指定してください。

```
@reboot $DTP_HOME/bin/cronguard.sh > $DTP_HOME/logs/cronguard.log 2>&1
```

トラブルシューティング

このセクションの内容：

- Report Center
- Cache Report Executor
- Team Server
- ソース表示機能
- デバッグ ログ モードへの切り替え

Cache Report Executor

デフォルトでは、staticLinksConfig.xml で指定されたレポートは午前 5 時にキャッシュされ始めます。午前 8 時までにキャッシュされないレポートがある場合、次の点を確認してください。

- レポート キャッシュがオンになっているか？

577 ページの「データベース接続の変更」および 206 ページの「Report Center データ キャッシュの有効化 / 無効化」を参照してください。

- CacheReportExecutor ジョブの allowFromHour の設定はどうなっているか？

CronConfig.xml ファイルを確認してください。

- 今朝 CacheReportExecutor は起動したか？

DTP_HOME\logs\recalculators.log を確認してください。このログ ファイルには Report Center のすべてのバックグラウンド ジョブが記録されます。

このファイルの最後に移動し、「CacheReportExecutor」を上に向かって検索します。

CacheReportExecutor は、staticLinksConfig.xml で指定されたレポートを実行するとき「Generating cache for」で始まるテキスト行をログに記録します。そのため、「Generating cache for」で検索することもできます。

- CacheReportExecutor は起動したかもしれないが、Report Center のレポートを参照するユーザーによって一時停止された。そして最後のユーザー アクションの後に待機時間が過ぎてまた動き出した可能性がある。

夜間にキャッシュする必要があるレポートが staticLinksConfig.xml で指定されていない場合、そのレポートを staticLinksConfig.xml に追加してください。詳細については 237 ページの「キャッシュするレポートの追加」を参照してください。

Team Server

問題： Team Server は起動するが、リクエストをまったく処理できない。

解決方法： デプロイが正しく実行されなかったことが考えられます。691 ページの「一般的な設定」のデプロイ操作をもう一度実行してください。

問題： Team Server は起動するが、接続の検証リクエストを除いて、すべてのリクエストでエラーメッセージを生成する。パスワードが不正か、または設定されていません。

解決方法： ライセンスの操作をもう一度実行してください。

問題： 使用している Jtest プラグインが最新かどうかを判断するには、どうすればいいのか？

解決方法： インストール ディレクトリの下に `\tomcat\webapps\` ディレクトリに移動し、`jtest` ディレクトリのタイムスタンプを確認します。`jtest` ディレクトリのタイムスタンプが `jtest.war` のタイムスタンプよりも古い場合、`jtest` ディレクトリを削除してサービスを再起動します。最新の Jtest プラグインが自動的に適用されます。

ソース表示機能

Report Center がファイル ソースを表示できない場合は、DTP_HOME/logs/rs.log を参照し、エラーの詳細メッセージを確認してください。

問題：クライアント コマンド (Cleartool、CCM など) にアクセスできないという内容のメッセージがある。原因の可能性として、クライアントがインストールされていないことが考えられる。

解決方法：クライアントをインストールし、コマンドラインからコマンドを発行してコマンドにアクセスできることを検証します。

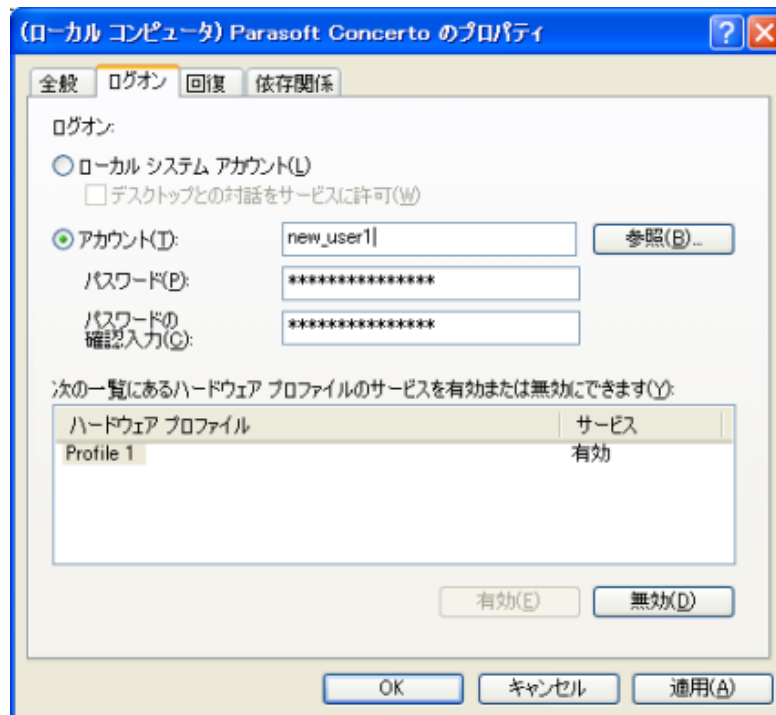
問題： クライアント コマンドラインは動作するが、Report Center がやはりコマンドにアクセスできない。原因の可能性として、DTP Server を実行するユーザーがクライアント コマンドにアクセスできないことが考えられる。

解決方法 1： DTP Server を実行するユーザーがクライアント コマンドにアクセスできるようにします。

解決方法 2： DTP Server を実行するユーザーを、クライアント コマンドにアクセスできるユーザーに変更します。次の操作を行います。

1. [コンピューターの管理] ウィンドウを表示して、使用するマシン上のサービスの一覧にアクセスします。
 - a. デスクトップの [マイ コンピューター] アイコンを右クリックします。
 - b. ショートカット メニューの [管理] をクリックします。
 - c. 左側のリストから [サービスとアプリケーション] > [サービス] を選択します。
2. Parasoft DTP サービスを右クリックし、ショートカット メニューの [プロパティ] をクリックします。

Parasoft DTP サービスのプロパティ ダイアログが表示されます。
3. [ログオン] タブをクリックします。
4. [アカウント] を選択し、適切なユーザーを入力または参照して指定します。



5. [適用] ボタンをクリックしてから [OK] をクリックします。
6. サービスを再起動します。

ClearCase のソース表示機能

問題 1:

Source Scanner が異なるサーバー上で実行される。その結果、Source Scanner と Report Center で ClearCase ビューのパスが異なる。Source Scanner は、ソース リポジトリをスキャンするときに次の情報を Report Center に送信する。

- ソース ファイル情報 (パス、リビジョンなど)
- ソース リポジトリへの接続パラメーター。ソース表示機能のために Report Center が使用する。

解決方法: ClearCase の場合、ソース リポジトリ パラメーターの 1 つはビューへのパスです。Source Scanner はこのパラメーターを内部的な CLEAR_CASE_PATH パラメーターとして送信します。

重要! Source Scanner が DTP と異なるマシンで実行され、ClearCase ビューへの異なるパスを持つ場合、DTP Server 用に ClearCase ビューのパスを指定する必要があります。

ClearCase ビューの場所を指定するには、RSConfig.xml ファイル中の次の行を編集します。

```
<clearcase-view-path>PUT_CLEARCASE_VIEW_LOCATION_HERE</clearcase-view-path>
```

問題 2:

ClearCase の動的ビューと静的ビュー (スナップショット ビュー) のサポート。

解決方法: Report Center のソース コード ビューワーは ClearCase の動的ビューと静的ビュー (スナップショット ビュー) の両方をサポートします。さらに、SourceScanner と Report Center は異なる種類のビューに対して動作できます。たとえば、SourceScanner がスナップショット ビューをスキャンし、Report Center がマシンで定義された動的ビューを持つことができます。

動的ビュー:

```
M:\my_dynamic_view\myvob\src\com\parasoft\MyFile.java
```

スナップショット ビュー:

```
C:\ClearCase\views\my_snapshot_view\myvob\src\com\parasoft\MyFile.java
```

Report Center での唯一の必要事項は、「問題 1:」で説明しているように、ビューへのパスを適切に定義することです。

デバッグ ログ モードへの切り替え

統合の問題や他の予想外の動作をトラブルシューティングするための手段として、DTP Server をデバッグ ログ モードに切り替えることができます。

1. DTP_HOME/bin/log_config.xml を編集して次のように変更し、DTP のログ レベルを DEBUG に設定します。

変更前 :

```
<root>
<priority value="WARN"/>
<appender-ref ref="FILE"/>
</root>
```

変更後 :

```
<root>
<priority value="DEBUG"/>
<appender-ref ref="FILE"/>
</root>
```

詳細については 763 ページの「.log ファイル プロパティの変更」を参照してください。

2. このステップは、デバッグ モデルの DTP ソース管理関連モジュールを有効化します。ソース管理に関連する機能で問題が発生した場合にだけ次の変更を行ってください。

DTP の Java VM 起動パラメーターに次のパラメーターを追加します。

```
-Dscontrol.log=true
-Dcom.parasoft.xtest.logging.config.jar.file=/com/parasoft/xtest/
logging/log4j/config/eclipse.on.xml
```

3. DTP に再びデータを送り、適切なログ情報を得るために動作の再現を試みます。
4. rs.log をチェックします。このファイルは通常の起動中に 50MB を超えることがあります。

索引