



Red Hat Developer Tools 2019.3

Eclipse の使用

アプリケーションを使用した Eclipse 4.12.0 のインストールと最初の手順

Red Hat Developer Tools 2019.3 Eclipse の使用

アプリケーションを使用した Eclipse 4.12.0 のインストールと最初の手順

Robin Owen
kowen@redhat.com

Yana Hontyk
yhontyk@redhat.com

法律上の通知

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

Red Hat Developer Tools のインストールおよび使用開始用のユーザー情報

目次

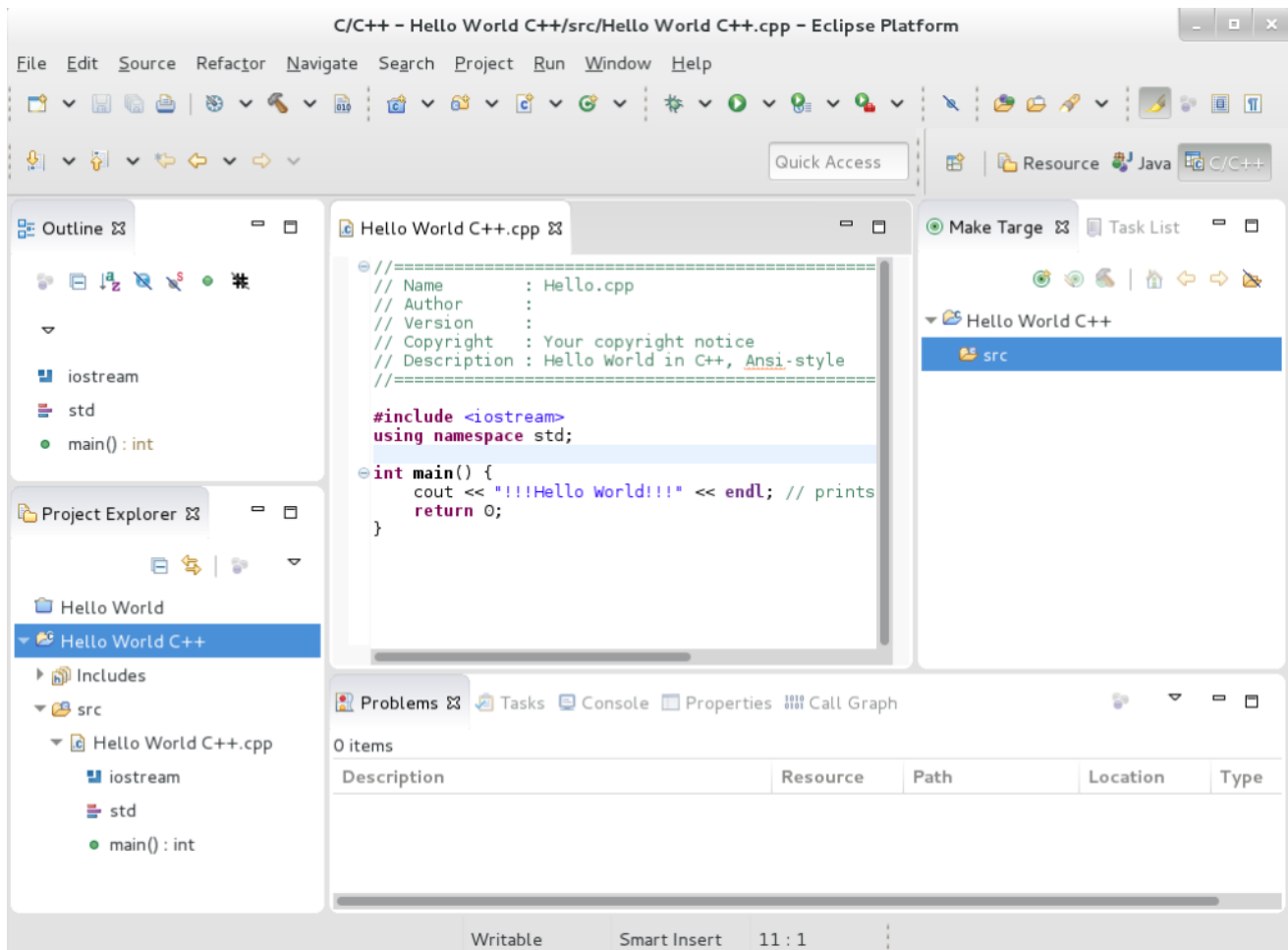
第1章 ECLIPSE 4.12.0	3
1.1. RED HAT DEVELOPER TOOLS リポジトリの有効化	5
1.2. ECLIPSE のインストール	6
1.3. ECLIPSE の使用	7
1.4. 関連資料	8
第2章 DEVELOPER TOOLS 2019.2 以降の変更点	10
第3章 ECLIPSE 4.12.0 の既知の問題	12
3.1. ECLIPSE が起動時にハングする	12
3.2. NULLPOINTEREXCEPTION とサードパーティーソースのプラグインがある	12
3.3. DEBUGINFO 競合	12
3.4. RHSCL RH-MAVEN33-SCLDEVEL と RH-MAVEN35-SCLDEVEL パッケージの競合	12
3.5. PYDEV ユーザーが PIP 統合で問題が発生する可能性がある	13
3.6. PYTHON モジュールを管理するための PYDEV ショートカットが機能しない	13
3.7. ECLIPSE SUBCLIPSE とベース RHEL SUBVERSION 間の非互換性	13
3.8. TYCHO 競合	13

第1章 ECLIPSE 4.12.0

Red Hat Enterprise Linux 7 で利用できる Red Hat Developer Tools には、Eclipse 4.12.0 が含まれています。これは、Eclipse Foundation の 2019-06 リリースへの依存に基づいています。**rh-eclipse** コレクションには、**rh-maven35** コレクションが必要なことに注意してください。

Eclipse は、開発プロセスの各フェーズにツールを提供する強力な開発環境です。さまざまな異種ツールを統一された環境に統合し、豊富な開発エクスペリエンスを構築し、完全に設定可能なユーザーインターフェースを提供し、さまざまな方法で拡張できるプラグ可能なアーキテクチャーを特長としています。たとえば、**Valgrind** プラグインを使用すると、プログラマーは、**Eclipse** ユーザーインターフェースを介してコマンドラインで実行されるメモリープロファイリングを実行できます。

図1.1 Eclipse セッションの例



Eclipse は、コマンドラインツールによる従来の対話に代わるグラフィカルな開発環境を提供します。コマンドラインインターフェースを使用しない開発者には代替手段となります。従来の、ほとんどがコマンドラインベースの Linux ツール (**gcc** または **gdb** など) と **Eclipse** は、プログラミングに 2 つのアプローチを提供します。

Red Hat JBoss Middleware のアプリケーション開発方法、または OpenShift Tools のサポート方法については、[Red Hat Developer Studio](#) を参照してください。

表1.1 rh-eclipse に含まれる Eclipse コンポーネント

パッケージ	説明
-------	----

パッケージ	説明
rh-eclipse-eclipse-cdt	CDT (C/C++ 開発ツールリング) は、これらの 2 つの言語で開発するための機能およびプラグインを提供します。
rh-eclipse-eclipse-changelog	changelog ファイルを作成および維持できる ChangeLog プラグイン。
rh-eclipse-eclipse-dtp	Eclipse Data Tools Platform プロジェクト
rh-eclipse-eclipse-egit	Eclipse のチームプロバイダーで、Git リポジトリと対話するための機能およびプラグインを提供する EGit。
rh-eclipse-eclipse-emf	EMF (Eclipse Modeling Framework) を使用すると、構造化データモデルに基づいてアプリケーションをビルドできます。
rh-eclipse-eclipse-epp-logging	Eclipse エラー報告ツール。
rh-eclipse-eclipse-gcov	GCov テスト対象プログラムを Eclipse と統合する GCov プラグイン。
rh-eclipse-eclipse-gef	既存のアプリケーションモデルからリッチグラフィカルエディターを作成できるようにする GEF (Graphical Editing Framework)。
rh-eclipse-eclipse-gprof	Gprof パフォーマンス分析ユーティリティを Eclipse と統合する Gprof プラグイン。
rh-eclipse-eclipse-jdt	Eclipse Java 開発ツール (JDT) プラグイン。
rh-eclipse-eclipse-jgit	JGit は、 Git リビジョン管理システムの Java 実装です。
rh-eclipse-eclipse-manpage	Eclipse の man ページを表示できる man ページプラグイン。
rh-eclipse-eclipse-mpc	Eclipse Marketplace クライアント。
rh-eclipse-eclipse-mylyn	Mylyn は、 Eclipse のタスク管理システムです。
rh-eclipse-eclipse-oprofile	OProfile を Eclipse と統合する OProfile プラグイン。

パッケージ	説明
rh-eclipse-eclipse-pde	Eclipse プラグインを開発するためのプラグイン開発環境。
rh-eclipse-eclipse-perf	perf ツールを Eclipse と統合する Perf プラグイン。
rh-eclipse-eclipse-ptp	同期したプロジェクトにサポートを提供する PTP プロジェクト のサポートを提供する PTP プロジェクトのサブセット。
rh-eclipse-eclipse-pydev	Eclipse 向けの Python IDE の完全な機能。
rh-eclipse-eclipse-remote	拡張可能な remote-services フレームワークを提供する Remote Services プラグイン。
rh-eclipse-eclipse-rpm-editor	RPM 仕様ファイルの維持を可能にする Eclipse Spec File Editor。
rh-eclipse-eclipse-rse	Eclipse からのリモートシステムと連携できるようにする RSE (Remote System Explorer) フレームワーク。
rh-eclipse-eclipse-systemtap	SystemTap と Eclipse を統合する SystemTap プラグイン。
rh-eclipse-eclipse-subclipse	Subclipse は、Eclipse のチームプロバイダーで、Subversion リポジトリと対話するための機能とプラグインを提供します。
rh-eclipse-eclipse-valgrind	Valgrind を Eclipse と統合する Valgrind プラグイン。
rh-eclipse-eclipse-webtools	Eclipse Webtools プラグイン。

1.1. RED HAT DEVELOPER TOOLS リポジトリの有効化

Eclipse をインストールするには、Red Hat Developer Tools リポジトリを有効にします。

- Red Hat Enterprise Linux Server で **rhel-7-server-devtools-rpms** リポジトリを有効にするか、Red Hat Enterprise Linux Workstation で **rhel-7-workstation-devtools-rpms** リポジトリを有効にします。
- Red Hat Enterprise Linux Server で **rhel-server-rhsc1-7-rpms** リポジトリを有効にするか、Red Hat Enterprise Linux Workstation で **rhel-workstation-rhsc1-7-rpms** リポジトリを有効にします。



注記

Red Hat Developer Tools は、**クライアント** または **ComputeNode** のバリエーションではサポートされていません。

Red Hat Developer Tools リポジトリを有効にするには、以下を実行します。

1. root ユーザーで次のコマンドを実行します。

```
# subscription-manager repos --enable rhel-7-server-devtools-rpms
# subscription-manager repos --enable rhel-server-rhsccl-7-rpms
```



注記

上記のコマンドでは、**Server** または **Workstation** を使用している Red Hat Enterprise Linux のバリエーションに応じて、コマンドは以下のようになります。

- **Workstation** バリエーション:

```
# subscription-manager repos --enable rhel-7-workstation-devtools-rpms
```

- **Server** バリエーション:

```
# subscription-manager repos --enable rhel-7-server-devtools-rpms
```

開発者は、Red Hat Enterprise Linux Server を使用して、最も幅広い開発ツールにアクセスすることを推奨します。

サブスクリプションの登録および割り当ての詳細は、[Red Hat Subscription Management の使用および設定](#)を参照してください。

1.1.1. Red Hat Developer Tools debuginfo リポジトリの有効化

Red Hat Developer Tools は、リポジトリに含まれるすべてのアーキテクチャー依存 RPM の **debuginfo** パッケージも提供します。これらのパッケージは、コアファイル分析や Eclipse 自体のデバッグに役立ちます。Red Hat Developer Tools **debuginfo** リポジトリを有効にするには、以下を実行します。

1. root ユーザーで次のコマンドを実行します。

```
# subscription-manager repos --enable rhel-7-server-devtools-debug-rpms
# subscription-manager repos --enable rhel-server-rhsccl-7-debug-rpms
```

debuginfo パッケージのインストール、理解、および使用の詳細は、[実行中のアプリケーションのデバッグ](#)を参照してください。

1.2. ECLIPSE のインストール

Eclipse 開発環境は、RPM パッケージのコレクションとして提供されます。

1. Eclipse をインストールするには、**root** ユーザーとして以下のコマンドを実行します。

```
# yum install rh-eclipse
```

- 利用可能なコンポーネントの一覧は、[表1.1 「rh-eclipse に含まれる Eclipse コンポーネント」](#) を参照してください。



注記

rh-eclipse は C、C++、および Java 開発を完全にサポートしますが、Fortran プログラミング言語はサポートされません。

1.3. ECLIPSE の使用

rh-eclipse を起動するには、次のコマンドを実行します。

1. **Applications > Programming > Red Hat Eclipse** の順にクリックするか、シェルプロンプトで以下を入力します。

```
$ scl enable rh-eclipse eclipse
```

2. 起動中に、**Eclipse** はプロジェクトを保管するディレクトリーである **ワークスペース** を選択するように求められます。デフォルトのオプションである **~/workspace/** を使用するか、**Browse** をクリックしてファイルシステムを閲覧し、カスタムディレクトリーを選択します。また、**デフォルトとしてこれを使用** を選択すると、再度このチェックボックスを選択して、**Eclipse** がこのダイアログボックスを次回の開発環境を実行するときに表示しないようにできます。
3. **OK** をクリックして選択を確定し、起動を続行します。

1.3.1. Red Hat Developer Toolset Toolchain の使用

GNU コンパイラコレクション および **binutils** Red Hat Developer Toolset のサポートで **rh-eclipse** を使用するには、以下を行います。

1. 「[Eclipse の使用](#)」の説明に従って、**devtoolset-8-toolchain** がインストールされ、アプリケーションが実行されていることを確認します。**rh-eclipse** コレクションはデフォルトで Red Hat Developer Toolset ツールチェーンを使用します。

システムに **devtoolset-8-toolchain** パッケージをインストールする方法の詳細は、[Red Hat Developer Toolset User Guide セクション 1.5](#) を参照してください。[Red Hat Developer Toolset のインストール](#)



重要

以前 Red Hat Enterprise Linux バージョンの **GNU コンパイラコレクション** で構築されたプロジェクトの以前のビルド結果をすべて破棄します。これには、**Eclipse** でプロジェクトを開き、メニューから **プロジェクト > Clean** を選択します。

1.3.2. Red Hat Enterprise Linux Toolchain の使用

Red Hat Enterprise Linux に配布されるツールチェーンの **rh-eclipse** サポートに使用するには、プロジェクトの設定を変更して、Red Hat Enterprise Linux システムバージョンの **gcc**、**g++**、および **as** の絶対パスを使用します。

現在のプロジェクトのツールの Red Hat Enterprise Linux システムバージョンを明示的に使用するよう **Eclipse** を設定するには、以下を実行します。

1. C/C++ パースペクティブで **Project > Properties** をクリックし、プロジェクトプロパティーを開きます。
2. ダイアログボックス左側のメニューで、**C/C++ Build > Settings** をクリックします。
3. **ツール設定** タブをクリックします。
4. C プロジェクトで作業している場合は、以下を行います。

- a. **GCC C Compiler** または **Cross GCC Compiler** を選択し、**Command** フィールドの値を以下のように変更します。

```
/usr/bin/gcc
```

- b. **GCC C Linker** または **Cross GCC Linker** を選択し、**Command** フィールドの値を以下のように変更します。

```
/usr/bin/gcc
```

- c. **GCC C Assembler** または **Cross GCC Assembler** を選択し、**Command** フィールドの値を以下のように変更します。

```
/usr/bin/as
```

5. C++ プロジェクトで作業している場合:

- a. **GCC C Compiler*** または ***Cross G Compiler** を選択し、**Command** フィールドの値を以下のように変更します。

```
/usr/bin/g++
```

- b. **GCC C Compiler** または **Cross GCC Compiler** を選択し、**Command** フィールドの値を以下のように変更します。

```
/usr/bin/gcc
```

- c. **GCC C Linker*** または ***Cross G Linker** を選択し、**Command** フィールドの値を以下のように変更します。

```
/usr/bin/g++
```

- d. **GCC C Assembler** または **Cross GCC Assembler** を選択し、**Command** フィールドの値を以下のように変更します。

```
/usr/bin/as
```

6. **OK** をクリックして設定の変更を保存します。

1.4. 関連資料

Eclipse とその機能に関する詳細な説明は、本書の対象外です。詳しい情報は、以下の資料を参照してください。

ヘルプ > ツール設定 > C/C++ Build > Settings

インストールされているドキュメント

- Eclipse には、統合機能およびツールごとに幅広いドキュメントを提供する組み込みヘルプシステムが含まれています。

関連項目

- [2章 Developer Tools 2019.2 以降の変更点](#) では、以前のリリースの Developer Tools に含まれる Eclipse 開発環境に対する選択された機能および改良のリストを提供します。
- ユーザーガイドの [Red Hat Developer Toolset](#) の章では、Red Hat Developer Toolset の概要と、システムにインストールする方法の詳細を説明します。
- ユーザーガイドの [GNU コンパイラコレクション\(GCC\)](#) の章では、コマンドラインの C、C++、および Fortran で書かれたプログラムをコンパイルする方法を説明します。

第2章 DEVELOPER TOOLS 2019.2 以降の変更点

Red Hat Developer Tools は、2019-06 リリースの Eclipse 4.12.0 およびその他のプラグインと共に配布されており、Red Hat Developer Tools 2019.2 で配布されるバージョンに対するバグ修正および機能拡張が数多く追加されました。以下は、本リリースにおける新機能および互換性の変更点の一覧です。これらの新機能の使用方法は、組み込み Eclipse ドキュメントを参照してください。

依存関係

rh-eclipse コレクションは、**rh-maven35** コレクションによって異なります。RHEL 7.6 は、最適なエクスペリエンスを得るために推奨されます。

重要なパッケージの更新

- **eclipse** 4.11.0 → 4.12.0: Eclipse Platform JDT/PDE プラグインが 4.12.0 に更新されました。主な機能は次のとおりです。
 - プロジェクトをコマンドラインパラメーターとして渡すことで、プロジェクトのインポートを許可します。
 - 起動設定は「Quick Access」(Ctrl+3) セレクターからアクセス可能になりました。
 - Java 12 のサポートが改善されました。これにより、プログラマーが新しい言語機能を簡単に活用できる新しい修正やコードテンプレートが数多く追加されました。
 - メソッド呼び出しおよびメソッド実装に関する情報で Java エディターを強化できる新しいコードマイニングオプション。
 - JPMS モジュール依存関係を設定するための新しい UI。
詳細は、アップストリームのリリースノート <https://www.eclipse.org/eclipse/news/4.12/> を参照してください。
- **eclipse-cdt** 9.7.0 → 9.8.0: CDT プラグインは 9.8.0 に更新され、静的な分析、リファクタリング、および C/C++ コードフォーマッターの改善が加えられました。
詳細は、アップストリームのリリースノート (<https://projects.eclipse.org/projects/tools.cdt/releases/9.8.0/bugs>) を参照してください。
- **eclipse-linuxtools** 7.2.0 → 7.3.0: Linuxtools プラグインが 7.3.0 に更新され、バグを修正し、Docker ツールを改善しました。
詳細は、アップストリームのリリースノート (<https://projects.eclipse.org/projects/tools.linuxtools/releases/7.3.0/bugs>) を参照してください。
- **eclipse-webtools** 3.13.0 → 3.14.0: Webtools プラグインは 3.14.0 に更新され、さまざまなバグを修正し、Java 12 ファスティングプロジェクトのサポートを追加しました。
詳細は、アップストリームのリリースノート (<https://projects.eclipse.org/projects/webtools/releases/3.14/bugs>) を参照してください。
- **eclipse-pydev** 7.1.0 → 7.2.1: Pydev プラグインは 7.2.1 に更新され、多くのデバッガーの改善や構文強調表示の修正が含まれます。
詳細は、アップストリームのリリースノート (http://www.pydev.org/history_pydev.html) を参照してください。
- **eclipse-egit/eclipse-jgit** 5.3.0 → 5.4.0: Git 統合プラグインが 5.4.0 に更新され、パフォーマンス向上を追加し、以下のようなバグを修正しました。

- 新しい Apache SSH クライアントを使用する際に、パスワードで保護された ed25519 キーに対する新規スタイルの OpenSSH 秘密鍵への対応
- **--ssh apache** オプションを使用したコマンドラインで新しい Apache SSH クライアントのサポート
さらに多数あります。

詳細は、アップストリームのリリースノート

(<https://projects.eclipse.org/projects/technology.jgit/releases/5.4.0> および <https://projects.eclipse.org/projects/technology.egit/releases/5.4.0>) を参照してください。

- **eclipse-ptp** 9.2.1 → 9.3.0: Parallel Tools Platform プラグインが 9.3.0 に更新され、Java 11+ との互換性が改善され、GEM モジュールおよび SCI モジュールが削除されています。

詳細は、アップストリームのリリースノート

(<https://projects.eclipse.org/projects/tools.ptp/releases/9.3.0/bugs>) を参照してください。

第3章 ECLIPSE 4.12.0 の既知の問題

本セクションでは、Eclipse 4.12.0 の既知の問題について説明します。

3.1. ECLIPSE が起動時にハングする

原因: スレッドデッドロックが原因で、Eclipse が起動時にハングする可能性があります。

結果: ワークスペース選択ダイアログを表示せずに Eclipse スプラッシュスクリーンのみが表示されます。

回避策: PID を特定し、プロセスの kill シグナルを送信することで、問題のある Eclipse プロセスを開始します。たとえば、以下のようになります。

```
$ jps
30066 Jps
31631 org.eclipse.equinox.launcher_1.5.300.v20190308-0658.jar
$ kill 31631
```

結果: 問題のあるプロセスが強制終了されると、Eclipse が正常に再起動できます。

3.2. NULLPOINTEREXCEPTION とサードパーティーソースのプラグインがある

サードパーティー更新サイトからのプラグインがインストールされると、Eclipse がワークスペースログファイル **NullPointerException** で開始できなくなることがあります。

回避策: **-clean** オプションを使用して Eclipse を再起動し、依存関係解決キャッシュを消去します。以下に例を示します。

```
$ scl enable rh-eclipse "eclipse -clean"
```

結果: Eclipse は通常どおりに起動します。

3.3. DEBUGINFO 競合

原因: ユーザーがこのコレクションの **debuginfo** パッケージをインストールしようとする時、たとえば **rh-eclipse-eclipse-cdt-debuginfo**、コンテンツが以前のコレクションにある同じパッケージと競合する可能性があります。

結果: **rh-eclipse48** コレクションから同じ **rh-eclipse debuginfo** パッケージをインストールすると、**debuginfo** パッケージのインストールが失敗する可能性があります。

回避策: **rh-eclipse** コレクションの **debuginfo** を使用するには、**yum remove rh-eclipse48*debuginfo** コマンドを使用して、**rh-eclipse48** コレクションの **debuginfo** パッケージを最初にアンインストールする必要があります。

結果: **rh-eclipse** コレクションからの **debuginfo** パッケージのインストールが正常に実行されました。

3.4. RHSCL RH-MAVEN33-SCLDEVEL と RH-MAVEN35-SCLDEVEL パッケージの競合

RHSC `rh-maven33-scldevel` と `rh-maven35-scldevel` パッケージに競合があります。これは `rh-eclipse-scldevel` パッケージに影響を及ぼします (デフォルトではこのパッケージがインストールされていないことに注意してください)。

`rh-eclipse-scldevel` および `rh-eclipse46-scldevel` パッケージを同時に存在させることはできません。

3.5. PYDEV ユーザーが PIP 統合で問題が発生する可能性がある

Python SCL によって事前にインストールされるモジュールはアンインストールできません。これを実行すると、パーミッション拒否エラーが発生します。

3.6. PYTHON モジュールを管理するための PYDEV ショートカットが機能しない

`Ctrl+2` ショートカットを使用して、`pip` を選択しても適切に機能しません。

結果: このショートカットの使用を試行すると、ワークスペースログで `SWTException` が作成されます。

回避策: ユーザーは、`PyDev > Interpreters` 設定を使用して Python モジュールを引き続き管理できます。

3.7. ECLIPSE SUBCLIPSE とベース RHEL SUBVERSION 間の非互換性

Eclipse Subclipse で作成した Subversion リポジトリのワーキングコピーは、Subversion のベース RHEL バージョンと互換性がありません。このような作業コピーで `svn` コマンドを使用すると、以下のエラーが発生する可能性があります。

```
$ svn up
svn: E155021: This client is too old to work with the working copy
```

回避策: コマンドラインで Eclipse Subclipse によって使用される Subversion の Pure-Java 実装を使用します。

```
# yum install rh-eclipse-svnkit-cli # Command line support for SVNKit
```

ここで、`jsvn` コマンドを通常使用する場所で `svn` を使用できます。

```
$ jsvn up
Updating '.':
At revision 16476.
```

3.8. TYCHO 競合

原因: `rh-eclipse-tycho` パッケージは、前述のコレクションから同じパッケージと競合します。例: `rh-eclipse48-tycho`

結果: `rh-eclipse48-tycho` パッケージがすでにインストールされていると、`rh-eclipse-tycho` パッケージのインストールが失敗する可能性があります。

回避策: Eclipse を構築または再ビルドするユーザーのみが、tycho を必要とすることに注意してください。必要に応じて、**rh-eclipse48-tycho** パッケージをインストールする前に、**yum remove rh-eclipse48-tycho** コマンドを使用して **rh-eclipse-tycho** パッケージをアンインストールします。

結果: **rh-eclipse-tycho** パッケージのインストールが正常に実行されました。