



Red Hat Enterprise Linux 7

移行計画ガイド

Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 における主な相違点

Red Hat Enterprise Linux 7 移行計画ガイド

Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 における主な相違点

法律上の通知

Copyright © 2023 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

本ガイドでは、Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 への変更の概要と、Red Hat Enterprise Linux 7 への移行を評価について説明します。

目次

第1章 アップグレード方法	3
1.1. RED HAT ENTERPRISE LINUX 6 からのアップグレード方法	3
第2章 主な変更点と移行で考慮すべき点	4
2.1. システム制限	4
2.2. インストールおよび起動	4
2.3. ファイルシステムレイアウト	13
2.4. システム管理	16
2.5. ファイルシステムのフォーマット	22
2.6. 物理ストレージ	23
2.7. ネットワーク	26
2.8. クラスタリングと高可用性	35
2.9. デスクトップ	36
2.10. 開発者ツール	38
2.11. セキュリティーおよびアクセス制御	39
第3章 パッケージ、機能、およびサポートの変更	43
3.1. 新しいパッケージ	43
3.2. パッケージの置換	45
3.3. 非推奨パッケージ	53
3.4. 削除済みパッケージ	54
3.5. 削除済みドライバー	75
3.6. 非推奨ドライバー	77
第4章 移行関連の CUSTOMER PORTAL LABS	79
Red Hat Enterprise Linux Upgrade Helper	79
Product Life Cycle Checker	79

第1章 アップグレード方法

インプレースアップグレードは、システムを RHEL の次のメジャーバージョンにアップグレードする方法です。この方法は、推奨され、サポートされています。

1.1. RED HAT ENTERPRISE LINUX 6 からのアップグレード方法

[RHEL 6 から RHEL 7 へのアップグレード](#) ガイドでは、RHEL 6 から RHEL 7 へのインプレースアップグレード手順を説明します。現在、対応しているインプレースアップグレードパスは RHEL 6.10 から RHEL 7.9 までです。

SAP HANA を使用している場合は、代わりに [How do I upgrade from RHEL 6 to RHEL 7 with SAP HANA](#) に従ってください。SAP HANA を使用した RHEL のアップグレードパスは異なる場合があることに注意してください。

RHEL 6 から RHEL 7 へのアップグレードプロセスは、以下の手順で設定されます。

1. Red Hat は、システムのアップグレードをサポートします。
2. 必要なりポジトリおよびパッケージをインストールし、対応していないパッケージを削除して、システムのアップグレードの準備を行います。
3. Preupgrade Assistant を使用して、アップグレードに影響を与える可能性がある問題をシステムで確認してください。
4. Red Hat Upgrade Tool を実行してシステムをアップグレードします。

第2章 主な変更点と移行で考慮すべき点

本章では、Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 への移行に影響する可能性がある主な変更点と機能について説明します。各セクションをよく読んで、Red Hat Enterprise Linux 7 へのアップグレードがシステムに与える影響について十分に理解するようにしてください。

2.1. システム制限

Red Hat Enterprise Linux でサポートされるシステム制限は、バージョン 6 とバージョン 7 で異なります。

Red Hat Enterprise Linux 7 では、1 GB 以上のディスク領域がインストールで必要となります。ただし、Red Hat では、すべてのサポート対象のアーキテクチャーで 5 GB 以上のディスク領域を推奨しています。

AMD64 または Intel 64 のシステム稼働には、1 GB 以上のメモリーが必要になります。Red Hat では、論理 CPU あたり 1 GB 以上のメモリーを推奨しています。AMD64 または Intel 64 のシステムでのサポートには、以下の上限が設定されています。

- 最大 3 TB のメモリー (理論的上限: 64 TB)
- 最大 160 の論理 CPU (理論的上限: 5120 の論理 CPU)

64 ビット Power システムの稼働には、2 GB 以上のメモリーが必要になります。サポートには以下の上限が設定されています。

- 最大 2 TB のメモリー (理論的上限: 64 TB)
- 最大 128 の論理 CPU (理論的上限: 2048 の論理 CPU)

IBM System z システムの稼働には、1 GB 以上のメモリーが必要になり、理論的には以下の上限までサポート可能です。

- 最大 3 TB のメモリー
- 最大 101 の論理 CPU

Red Hat Enterprise Linux 7 の要件および制限に関する最新情報

は、<https://access.redhat.com/site/articles/rhel-limits> でオンラインで利用できます。ハードウェアまたはソフトウェアが認定されているかどうかを確認するには、<https://access.redhat.com/certifications> を参照してください。

2.2. インストールおよび起動

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間になされたインストールツールおよびプロセスの変更の概要について説明します。

2.2.1. 新ブートローダー

Red Hat Enterprise Linux 7 では、GRUB2 ブートローダーが導入されています。これは、Red Hat Enterprise Linux 7.0 およびそれ以降でのレガシー GRUB に代わるものです。GRUB2 は、以前のものよりも多くのファイルシステムと仮想ブロックデバイスをサポートします。利用可能なオペレーティングシステムを自動的にスキャンして、その設定を行います。ユーザーインターフェイスも改善され、ユーザーはブートローダーのインストールを省略することができます。

ただし、GRUB2 への移行により、MBR 形式のパーティションテーブルを持つ BIOS マシンでフォーマットされたパーティションに、ブートローダーをインストールすることができなくなります。一部のファイルシステムでコアブートローダーイメージの一部を移動する最適化機能が自動化され、GRUB レガシーブートローダーが破壊されることがあるため、この動作が変更になりました。GRUB2 では、ブートローダーが、MBR (Master Boot Record) 形式のパーティションテーブルを持つ BIOS マシンのパーティションテーブルと、最初のパーティションとの間で利用可能な領域にインストールされます。GPT (GUID Partition Table) 形式のパーティションテーブルを持つ BIOS マシンでは、ブートローダー用に特別な BIOS 起動パーティションを作成する必要があります。UEFI マシンでは、引き続きブートローダーを EFI システムパーティションにインストールできます。

ブートローダーが新しくなったため、推奨されるパーティションの最小サイズが変更しました。表 2.1「推奨される最小パーティションサイズ」では、新しい推奨事項の概要を説明します。詳細については、[MBR および GPT に関する考慮事項](#)を参照してください。

表2.1 推奨される最小パーティションサイズ

パーティション	BIOS & MBR	BIOS & GPT	UEFI & GPT
/boot	500 MB	/	10 GB
swap	RAM の 2 倍以上。詳細については、 推奨されるパーティションスキーム を参照してください。	ブートローダー	該当なし (パーティションテーブルと最初のパーティションの間にインストールされる)

ファイルシステムが破損するというリスクを承知の上で、**force** オプションを使用して手動で GRUB2 をフォーマット済みのパーティションにインストールするか、別のブートローダーを使用します。代替のブートローダーのリストについては、[インストールガイド](#)を参照してください。

デュアルブートシステムがある場合は、GRUB2 のオペレーティングシステム検出を使用して、どちらのオペレーティングシステムも起動できる設定ファイルを自動的に書き込みます。

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

重要

UEFI の使用に基づいているデュアルブートがある場合は、MBR レガシーベース以外のメカニズムを使用することに注意してください。EFI 固有の grub2 コマンドを使用する必要がないことを示しています。

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

2.2.1.1. デバッグに使用するデフォルトのブートエントリー

systemd のデフォルトのブートエントリーが `/etc/grub.cfg` ファイルに追加されました。手動でデバッグを有効にする必要はありません。デフォルトのブートエントリーを使用すれば、ブート時にオプションに影響を与えずにシステムをデバッグすることができます。

2.2.2. 新 Init システム

systemd は、Red Hat Enterprise Linux の以前のリリースで使用されていた SysV init システムに代わるシステムおよびサービスマネジャーです。

systemd は、ブート時に最初に開始し、シャットダウン時に最後に終了するプロセスです。残りのブートプロセスを調整し、そのユーザー向けにシステムを設定します。**systemd** では、独立したプログラムが並列して読み込めるので、ブートプロセスが格段に速くなります。

systemd は、ユーザーエクスペリエンスおよびスクリプト API の大部分に関して、SysV と互換性があります。しかし、例外もいくつかあります。詳細は、「[後方互換性](#)」を参照してください。

systemd の移行には、Red Hat Enterprise Linux 用の管理ツールの変更も関わってきます。詳細については、**systemctl** man ページまたは[システム管理者ガイド](#)を参照してください。

起動プロセスの詳細については、[インストールガイド](#)を参照してください。**systemd** の詳細については、[システム管理者ガイド](#)を参照してください。

2.2.2.1. 後方互換性

systemd は、ユーザーエクスペリエンスおよびスクリプト API においてほぼ SysV と互換性があります。しかし、互換性が限定的なケースもいくつかあります。

- 標準の `/etc/init.d/servicename` コマンド (**start**、**stop**、**status**) は依然として機能します。ただし、レガシーの `init` スクリプトを使用するのではなく **systemd** に直接転送するため、Red Hat では、`/usr/sbin/service servicename` コマンドを推奨します。
- ランレベルサポートは制限されています。すべての SysV ランレベルは **systemd** ターゲットにマッピングしますが、すべての **systemd** ターゲットが SysV ランレベルにマッピングするわけではありません。このため、現在のランレベルに対するチェックのうち、**N** (不明のランレベル) を返すものもあります。Red Hat では、ランレベルチェックを避けて、より有用な **systemd** ターゲットに移動することを推奨しています。
- レガシーランレベル 2、3、および 4 はすべてデフォルトで **multi-user.target** **systemd** ターゲットにマッピングを行います。この動作は、別の **systemd** ターゲットを設定すれば修正できます。
- サービスはクリーンに実行し、呼び出しているユーザーのコンテキストは継承されません。継承されるコンテキストに依存する `init` スクリプトは機能しません。
- **systemd** は、`init` スクリプト内で他の動詞をサポートしません。**start**、**stop**、または **status** 以外の動詞が必要な場合は補助スクリプトに移動してください。
- Linux Standard Base ヘッダー情報は、ランタイム時に **systemd** が完全に解釈、利用します。
- `init` スクリプト操作はすべて 5 分でタイムアウトするようになっており、`init` スクリプトのハングでシステムがフリーズすることを防ぎます。
- **systemd** は、稼働中のサービスだけを停止します。開始していないサービスがシャットダウン中に停止されることもありません。
- **chkconfig** ツールは、SysV サービスおよびランレベル情報のみを表示し、誤解を招く情報を出力する可能性があります。Red Hat では、代わりに **systemctl** コマンドの使用を推奨しています。
- **CPUAccounting** オプションが有効になっていると、`root` 権限を持つ SysV サービスであってもリアルタイムスケジューリングを取得できません。任意のサービスに **CPUAccounting** を有効化することで、**systemd** は CGroup CPU 帯域幅コントローラーをグローバルに利用して、後続の `sched_setscheduler()` システム呼び出しはリアルタイムスケジューリングの優先度によって予期せずに終了します。このエラーを避けるには、CGroup `cpu.rt_runtime_us` オプションは、サービスを使用してリアルタイムに設定できます。

- サービスは、標準入力 (stdin) から読み込むことはありません。インタラクティブなスクリプトが必要な場合は、**systemd** がサポートする最小パスワードクエリーフレームワークを検討してください。この機能の詳細は、man ページで入手できます。

```
$ man systemd-ask-password
```

- 以前のバージョンの Red Hat Enterprise Linux には System z 固有のプレインストールスクリプト (**linuxrc.s390**) が含まれており、これがシステムの起動時に System z システムを開始していました。新しい init システムではこのプレインストールが廃止され、System z システムは、AMD64、Intel 64、および Power システムと同じ方法でブートします。

2.2.2.2. Systemd-debug-generator

systemd-debug-generator は、ランタイムデバッグシェルを有効にし、起動時に特定のユニットをマスクするためのジェネレーターです。これは、カーネルコマンドラインを読み取り、次の3つのオプションを使用します。

systemd.mask=

このオプションを指定し、その後にユニット名を追加すると、実行時にそのユニットがマスクされます。システムのブート時に、システムの起動をデバッグする初期ブートトランザクションから特定のユニットを削除するのに便利です。このオプションは複数回指定できます。

systemd.wants=

このオプションを指定し、その後にユニット名を追加すると、そのユニットの開始ジョブが初期トランザクションに追加されます。これは、システムの起動時にユニットを追加する場合に便利です。このオプションは複数回指定できます。

systemd.debug_shell

このオプションを指定すると、**debug-shell.service** がブートトランザクションに追加されます。これにより、システムの起動時の早い段階で、tty9 にデバッグシェルが発生します。シェルは、**systemctl enable** コマンドを使用して永続的に有効にすることもできます。

2.2.2.3. 新インストーラー

Red Hat Enterprise Linux のインストーラーである Anaconda は、Red Hat Enterprise Linux 7 のインストールプロセスを改善するために再設計、機能強化されました。

インストーラーの更新された機能は、以下のとおりです。

- グラフィカルユーザーインターフェイスが再設計され、より速く柔軟性が高くなり、ユーザーからの入力が少なく済みます。
- LVM シンプロビジョニングをサポート。
- btrfs のインストールサポート。(ただし、btrfs は Red Hat Enterprise Linux 7 ではテクノロジープレビューであることに注意してください)。
- ローカリゼーションサポートの改善。
- 直接フォーマット、パーティション化されないデバイスをサポート。
- ネットワークテクノロジーのチーミングおよびボンディングをサポート。
- 適切なキーボードのレイアウト、言語、およびタイムゾーンの自動選択をサポート (インターネット接続が必要)。(これにはインターネット接続が必要です。) 検出に基づいて設定された値は、手動で設定した値で上書きされます。

- DHCP が通知する NTP サーバーが自動的に使用されるようになりました。
- **realmd** DBus サービス、Active Directory、および FreeIPA 向けにキックスタートを統合。
- IBM System z および PowerPC システム、およびシリアルコンソールで機能する新テキストモード。テキストモードは、グラフィカルインストーラーが提供する機能のサブセットを提供します。

この新インストーラーに関する重要な変更もいくつかあります。

- これまでは、ユーザーはストレージ設定の際に、使用するストレージシステムに関する詳細な技術的知識が必要でした。Red Hat Enterprise Linux 7 では、ストレージ設定が再設計されているので、ユーザーは最小限の詳細を入力するだけで済みます。
- Anaconda は、**inst.root** パラメーターではなく、**inst.repo** パラメーターを使ってネットワークやその他のインストールの場所を設定します。
- グラフィカルインストーラーインターフェイスの詳細なパッケージ選択に替わって、**ソフトウェア選択** 画面が使用されるようになりました。ソフトウェアは **環境** と **アドオン** に分かれており、環境から1つ、そしてアドオンは好きなだけ選択できます。ユーザーは、1つの環境と任意の数のアドオンを選択します。キックスタートインストールでは、インストール時に選択したパッケージを完全に制御できます。

これらの機能の詳細については、[インストールガイド](#)を参照してください。

2.2.2.4. ブートパラメーターの変更

2.2.2.4.1. ブートパッケージの指定

インストーラーに固有のブートオプションには、本書では接頭辞 **inst.** が付いています。現在、Red Hat Enterprise Linux 7 ではこの接頭辞はオプションです。**resolution=1024x768** の機能と、**inst.resolution=1024x768** の機能は完全に同一です。ただし、今後のリリースではこの接頭辞は必須になると予想され、接頭辞がないパラメーターは廃止されたものとみなされます。

2.2.2.4.2. ブートパラメーターへの変更

新インストーラーでは、dracut を使用してディスクおよびネットワークを設定します。したがって、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 の間で、カーネルコマンドラインのブートパラメーターにいくつか変更がなされました。

新パラメーター

inst.stage2

読み込み対象のインストールプログラムのランタイムイメージの場所を指定します。構文は、**inst.repo** パラメーターと同じです。このオプションはイメージ以外のものをすべて無視し、パッケージ場所の指定に使用することはできません。

inst.dd

指定された場所にあるパッケージでドライバーパッケージを更新します。このオプションは複数回使用できます。場所に関する構文は、**inst.repo** パラメーターと同じです。

inst.geoloc

言語およびタイムゾーンを事前設定するために、インストーラーにどの位置情報を使用するかを設定します。デフォルト値は **provider_fedora_geopip** です。このパラメーターで有効な値には、以下のものが含まれます。

表2.2 位置情報の値

値	Effect
0	位置情報を無効にします。
provider_fedora_geoup	Fedora GeolP API を使用します。
provider_hostip	Hostip.info GeolP API を使用します。

inst.usefbx

ハードウェア固有のドライバーではなく、フレームバッファXドライバーの使用を指定します。このオプションは、**inst.xdriver=fbdev** と同じです。

bootdev

起動インターフェイスを指定します。このオプションは、**ip** を 2 回以上指定する場合に必須となります。

inst.multilib

multilib パッケージ用にシステムを設定し、たとえば、64 ビットシステム上への 32 ビットパッケージのインストールを可能にします。

gpt

マスターブートレコード (MBR) ではなく、GUID パーティションテーブル (GPT) にパーティション情報をインストールします。

inst.virtio

ログ転送に使用する virtio ポートを指定します。デフォルト値は、**org.fedoraproject.anaconda.log.0** です。このポートが存在する場合は、それが使用されません。

rd.dasd

DASD (ダイレクトアクセスストレージデバイス) アダプターデバイスバス識別子を取り、オプションでコンマ区切りの **sysfs** パラメーターと値のペアをとります。指定されたデバイスバス ID で DASD をアクティベートし、**sysfs** パラメーターを指定された値に設定します。たとえば、**rd.dasd=adaptor_id,readonly=0** となります。このパラメーターは、複数の DASD をアクティベートするために、複数回にわたり指定することができます。

rd.zfcp

FCP (zFCP) 上の SCSI アダプターデバイスバス識別子、WWPN (ワールドワイドポート名前)、および FCP LUN をとります。指定されたデバイスバス識別子、ポート名前、および LUN で zFCP デバイスをアクティベートします。複数の zFCP デバイスをアクティベートするには、このパラメーターを複数回にわたり指定することができます。

```
rd.zfcp=0.0.4000,0x5005076300C213e9,0x5022000000000000
```

rd.znet

ネットワークプロトコルタイプおよびコンマ区切りのサブチャンネルのリストと、必要に応じてコンマ区切りの **sysfs** パラメーターと値のペアをとります。指定されたプロトコル用に System z ネット

ネットワークデバイスドライバをアクティベートし、指定したサブチャンネルを設定し、指定されたとおりにパラメーターを設定します。複数のネットワークデバイスをアクティベートするには、このパラメーターを複数回にわたり指定することができます。

```
rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1,portname=foo
rd.znet=ctc,0.0.0600,0.0.0601,protocol=bar
```

変更されたパラメーター

inst.ks.sendmac

以前は **kssendmac** でした。発信 HTTP リクエストにヘッダーを追加します (全ネットワークインターフェイスの MAC アドレスを含みます)。これは、システムのプロビジョンに **inst.ks=http** を使用する際に有用です。

ネームサーバー

以前は **dns** でした。ネームサーバーのアドレスを指定します。このオプションは複数回使用できません。

非推奨パラメーター

このリストにあるオプションは推奨されません。動作はしますが同じ機能を提供している別のオプションがあります。廃止予定のオプションの使用は推奨されせん。今後のリリースで削除される予定です。

updates

インストールプログラムの更新の場所を指定するときに使用していました。 **inst.updates** オプションを指定します。

method

インストール方法の設定に使用されていました。代わりに **inst.repo=** オプションを使用してください。

repo

NFS インストールで、ターゲットがインストール可能なツリーではなく、NFS サーバー上にある ISO イメージであることを指定。この違いは自動的に検出されるようになっているため、このオプションは **inst.repo=nfs:server:/path** と同じになります。

dns

ドメインネームサーバー (DNS) の設定に使用していました。代わりに **nameserver=** オプションを使用してください。

netmask、gateway、hostname、ip、ipv6

これらのオプションは、**ip** オプションに統合されました。

ip=bootif

PXE サーバーからインストール時に使用される **BOOTIF** オプションを指定。これは自動検出されるようになっています。

ksdevice

キックスタートインストール中に使用するネットワークデバイスを設定。以下の表にあるように、このパラメーターの値は、それぞれ別のパラメーターに置き換えられています。

表2.3 キックスタートパラメーターの値

値	現在の動作
存在しない場合	デバイスや設定が ip または BOOTIF オプションで指定されていない限り、DHCP ですべてのデバイスの有効化を試行します。
ksdevice=link	無視 (デフォルト動作と同じ)
ksdevice=bootif	無視 (指定されている場合は BOOTIF がデフォルトとして使用されます)。
ksdevice=ibft	dracut のオプション ip=ibft に置き換え
ksdevice=MAC	BOOTIF=MAC で置き換えられます。
ksdevice=device	dracut ip オプションでデバイス仕様で置換えられます。

blacklist

指定したドライバーの無効化に使用していました。これは、以下の構文で **rd.driver.blacklist** **dracut** オプションで処理されるようになりました。

```
rd.driver.blacklist=mod1,mod2,...
```

nofirewire

FireWire インターフェイスのサポートの無効化に使用していました。代わりに **rd.driver.blacklist** オプションを使用して、FireWire ドライバー (**firewire_ohci**) を無効にすることができます。

```
rd.driver.blacklist=firewire_ohci
```

削除済みパラメーター

以下のオプションは削除されました。ここに挙げるオプションは、以前の Red Hat Enterprise Linux リリースでは使用されていましたが、現行リリースでは使用することができません。

serial

出力に **/dev/ttyS0** コンソールを使用するよう Anaconda に強制するために使用されていました。代わりに、**console** パラメーターを使用して **/dev/ttyS0** コンソール (もしくは同様のもの) を指定してください。

essid、wepkey、wpakey

ワイヤレスのネットワークアクセスを設定する際に使用していました。ネットワーク設定は、**dracut** が処理するようになりました。これはワイヤレスネットワークをサポートしないので、これらのオプションが使用不能になります。

ethtool

低レベルのネットワーク設定に使用していました。ネットワーク設定はすべて、**ip** オプションで処理されています。

gdb

ローダーのデバッグを許可する場合に使用していましたが、代わりに **rd.debug** を使用してください。

inst.mediacheck

インストール開始前のインストールメディアの検証に使用していましたが、**rd.live.check** オプションに置き換えられました。

ks=floppy

フロッピーディスクを Kickstart ファイルソースとして指定。フロッピードライブはブートメディアとしてサポートされなくなりました。

display

リモートディスプレイの設定に使用していましたが、**inst.vnc** オプションに置き換えられました。

utf8

テキストモードでのインストール時に UTF8 サポートを追加。UTF8 サポートは自動的に機能するようになりました。

noipv6

インストールプログラムの IPv6 サポートを無効化。IPv6 はカーネルに組み込まれたので、ドライバーがブラックリストに載ることはありません。ただし、**ipv6.disable dracut** オプションを使って IPv6 を無効にすることは可能です。

upgradeany

アップグレードは Red Hat Enterprise Linux 7 で変更されました。詳細については、[1章 アップグレード方法](#)、[「Preupgrade Assistant」](#)、および [「Red Hat Upgrade Tool」](#) を参照してください。

vlanid

VLAN デバイスを設定。**dracut vlan** オプションに置き換えられます。

2.2.3. firstboot 実装への変更

Red Hat Enterprise Linux 7 では、**firstboot** が初期設定ユーティリティである **initial-setup** に置き換えられ、新インストーラーとの相互運用性が高まっています。基本的な **firstboot** の機能はインストーラーと **initial-setup** に移されました。

firstboot 向けに作成されたサードパーティー製のモジュールは、引き続き Red Hat Enterprise Linux 7 で動作します。ただし、**firstboot** は、将来のリリースで非推奨になる予定です。したがって、サードパーティー製モジュールのメンテナーは、インストーラーまたは初期セットアップツール向けのモジュールの更新を考慮する必要があります。

2.2.4. 起動時のマウント動作の変更

初期のバージョンの Red Hat Enterprise Linux は、**/etc/fstab** で指定されたすべてのパーティションをマウントできるかどうかに関わらず起動していました。この結果、必要なパーティションがなくてもシステムは起動し、問題なく起動したように見ることがありました。

この状況を防ぐために、Red Hat Enterprise Linux 7 では、**/etc/fstab** で定義されたパーティションを起動時にマウントできないと、起動が失敗します。パーティションがマウントできないときに、それが原因で起動が失敗することを防ぐには、**/etc/fstab** で新しい **nofail** パラメーターを使用します。

```
/dev/critical /critical xfs defaults 1 2
/dev/optional /optional xfs defaults,nofail 1 2
```

この例では、**/optional** でマウントするデバイスが正常にマウントできない場合に、それが原因で起動が失敗することはありません。

2.2.5. /etc/issue ファイルへの変更

Red Hat Enterprise Linux の以前のバージョンでは、製品名およびマシンのリリース番号が、**/etc/issue** ファイルに含まれていました。Red Hat Enterprise Linux 7以降、製品名とリリース番号は **/etc/os-release** に移動し、**/etc/issue** の最初の行には、**agetty** エスケープコード **\S** が含まれるようになりました。**\S** エスケープコードは、製品名、およびマシンのリリース番号を表示するコンソールで展開します。このコードは、**/etc/os-release** ファイルで定義される **PRETTY_NAME** 変数で示されます。



重要

\S エスケープコードの拡張は、コンソールからのみ有効です。これに対応していない環境で拡張を使用すると、**\S** だけが出力されます。

\S の詳細情報は、man ページの **agetty** を参照してください。

2.3. ファイルシステムレイアウト

Red Hat Enterprise Linux 7 では、ファイルシステムのレイアウトに、大きな変更が 2 つ加えられています。

- ディレクトリー **/bin**、**/sbin**、**/lib**、および **/lib64** は、**/usr** 下に移動しています。
- **/tmp** ディレクトリーは、一時ファイルストレージシステム (**tmpfs**) として使うことができるようになりました。
- **/run** ディレクトリーは、一時ファイルストレージシステム (**tmpfs**) として使用されるようになりました。アプリケーションは、**/var/run** を使用するのと同じように、**/run** を使用できるようになりました。

2.3.1. root ファイルシステムの新レイアウト

従来は、最低限必要なコンテンツのみを **/bin** および **/lib** ディレクトリー配下に置くことで、ブートプロセスが遅くなることを回避してきました。ユーティリティーのなかには、**/usr** パーティションをマウントするために、**root (/)** レベルに置かれる必要があるものもありました。これにより、他のユーティリティーが、複数レベルのディレクトリーにコンテンツを広げてしまうという状況になりました。たとえば、**/bin** と **/usr/bin** の両方にといったようになります。

Red Hat Enterprise Linux 7 では、**/bin**、**/sbin**、**/lib**、および **/lib64** のディレクトリーが、**/usr** に移動しています。**/usr** ファイルシステムは、ユーティリティーではなく **initramfs** により **root** レベルのディレクトリーにマウントできるので、パッケージコンテンツを 2 つの異なるディレクトリーレベルに分ける必要はなくなりました。このため規模が非常に小さい **root** ファイルシステムが可能となり、システムがディスク領域の共有をより効率的に行い、メンテナンスが容易になると同時に柔軟性と安全性が高まりました。

この変更の影響を受けないように、以前の **/bin** ディレクトリーは **/usr/bin** へのシンボリックリンクに、そして **/sbin** は **/usr/sbin** へのシンボリックリンクに変更になりました。

2.3.1.1. ファイルシステムのアップグレード準備



警告

`/usr` が別のパーティションにある場合はインプレースアップグレードができません。別のパーティションからの `/usr` の移動は、お客様の責任のもとで行ってください。

`/var` が別のパーティションにある場合は、手動で `/var/run` と `/var/lock` をシンボリックリンクに変換する必要があります。

```
# mv -f /var/run /var/run.runmove~
# ln -sfn ../run /var/run
# mv -f /var/lock /var/lock.lockmove~
# ln -sfn ../run/lock /var/lock
```



重要

パーティション設定スキームに関する `preupgrade-assistant` の結果にすべて対応してください。

準備が完了したら、[インストールガイド](#)を参照して、アップグレードプロセスの実行に関する詳細を確認してください。

2.3.1.2. アップグレード成功の確認

アップグレードプロセスの実行後に、アップグレードが予想通りに機能したかを確認することが重要になります。

- 以下のシンボリックリンクが存在するかを確認します。
 - `/bin` は `/usr/bin` へのシンボリックリンクです。
 - `/sbin` は `/usr/sbin` へのシンボリックリンクです。
 - `/lib` は `/usr/lib` へのシンボリックリンクです。
 - `/lib64` は `/usr/lib64` へのシンボリックリンクです。
 - `/var/run` は `/run` へのシンボリックリンクです。
 - `/var/lock` は `/run/lock` へのシンボリックリンクです。
上記のディレクトリーが想定どおりにシンボリックリンクである場合、さらに2つのチェックが必要になります。
- 以下の `find` コマンドの出力をチェックします。

```
# find /usr/{lib,lib64,bin,sbin} -name '.usrmove'
```

このコマンドにより表示されるファイルもしくはディレクトリーは、同じ名前のものがすでに `/usr` にあるため、`/usr` にコピーすることはできません。この命名に関する競合は、手動で解決する必要があります。

3. 保管しておきたいファイルについて、以下のディレクトリーをチェックします。

- `/var/run.runmove~`
- `/var/lock.lockmove~`

上記のディレクトリーがいずれもシンボリックリンクではない場合は、「[失敗したアップグレードからのリカバリー](#)」に示されるリカバリープロセスを実行する必要があります。

2.3.1.3. 失敗したアップグレードからのリカバリー

アップグレードプロセスが失敗する理由はいくつもあります。以下のコマンドの出力をチェックして、失敗した原因を確認してください。

```
# dmesg
# journalctl -ab --full
```

エラーが見つからない場合は、以下をチェックします。

- `/` が書き込み可能か
- `/usr` が書き込み可能か
- `/` に十分なスペースがあるか
- `/usr` に十分なスペースがあるか
- `/var` が `rhelup` ツールにマウントされているか

さらにヘルプが必要な場合は、Red Hat サポートにご連絡ください。

2.3.2. `/tmp` ディレクトリーへの移動

Red Hat Enterprise Linux 7 では、`/tmp` を一時ファイルストレージシステム (`tmpfs`) 用のマウントポイントとして使うことができます。

これを有効にすると、この一時的なストレージはマウントされたファイルシステムのように表示されますが、コンテンツの保管先は永続的なストレージデバイスではなく、揮発性メモリーになります。メモリーが不足している場合を除いて、`/tmp` 内のファイルがハードドライブに保管されることはありません。メモリーが不足している場合は、`swap` 領域が使用されます。つまり、`/tmp` のコンテンツは再起動すると持続しないこととなります。

この機能を有効にするには、以下のコマンドを実行します。

```
# systemctl enable tmp.mount
```

この機能を無効にするには、以下のコマンドを実行します。

```
# systemctl disable tmp.mount
```

Red Hat では、Red Hat Enterprise Linux 7 で使用される様々なタイプの一時ストレージスペースに、以下のものを利用することを推奨しています。

- デーモンなどの権限付きプロセスでは、`/run/processname` を使って一時データを保存。

- 大量のデータを保存するプロセス、もしくは再起動後も存続する一時データを必要とするプロセスには、`/var/tmp` を使用。
- その他のプロセスには、`/tmp` を使用して一時データを保存。

2.3.3. /run ディレクトリーへの移動



重要

Preupgrade Assistant は、初期リリースの Red Hat Enterprise Linux 7.0 でのこの変更の効果をチェックしませんでした。この問題は RHBA-2014:1627 で修正されました (<https://rhn.redhat.com/errata/RHBA-2014-1627.html>)。

Red Hat Enterprise Linux の前のバージョンでは、一部のプログラムで、起動初期に `/var` ディレクトリーをマウントする前に、実行時データを `/dev` ディレクトリーに格納できました。主な Linux ディストリビューションでは、`/dev` ディレクトリーはデバイスノードにのみ使用し、`/run` を代わりに使用することが推奨されています。

したがって、Red Hat Enterprise Linux 7 では、`/run` ディレクトリーは、`/var/run` ディレクトリーをバインドマウントする一時ファイルストレージシステム (`tmpfs`) です。同様に、`/run/lock` ディレクトリーは `/var/lock` ディレクトリーをバインドマウントするようになりました。`/run` と `/run/lock` に格納されたファイルは、永続的ではなくなり、再起動後に保持されません。つまり、アプリケーションはインストール時に 1 回実行するのではなく、起動時に独自のファイルおよびディレクトリーを再作成する必要があります。これには、`/etc/app_name` ディレクトリーが適しています。

起動時にファイルとディレクトリーを再作成する方法については、`tmpfiles.d` の man ページである `man tmpfiles.d` を参照してください。設定例については、`/etc/tmpfiles.d` にある設定例を参照してください。

2.4. システム管理

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされたシステム管理ツールおよびプロセスの変更の概要について説明しています。

2.4.1. デフォルトのプロセス最大数 (ulimit)

Red Hat Enterprise Linux 6 において、`root` 以外のユーザーでは、1 つの PAM セッションあたりのプロセスが 1024 個に制限されていました。Red Hat Enterprise Linux 7 では、デフォルトで、1 つの PAM セッションあたりのプロセスが 4096 個に増えました。

このデフォルト値は `/etc/security/limits.d/*-nproc.conf` ファイル (通常は、Red Hat Enterprise Linux 7 上の `/etc/security/limits.d/20-nproc.conf`) で指定します。このファイルが存在しない場合、`root` 以外のユーザーが所有できるプロセスの最大数は、Red Hat Enterprise Linux で `nproc (ulimit -u)` のデフォルト値を決定するものは何ですか? で説明されているように、プログラムで決定されます。

`root` 以外のユーザーが、現在の、1 つの PAM セッションあたりに利用できるプロセスの数は、`ulimit -u` コマンドを実行して確認できます。

2.4.2. 設定ファイルの構文

Red Hat Enterprise Linux 6 では、設定ファイル内に定義した値のエクスポートに `export` コマンドが使われていました。`export` コマンドを使用しない変数はエクスポートされず、対応する `init` スクリプト用の設定値としてのみ使用されました。以下は、`/etc/sysconfig/ssh` ファイルの例です。

```
AUTOCREATE_SERVER_KEYS=YES
export SSH_USE_STRONG_RNG=1
export OPENSSL_DISABLE_AES_NI=1
```

Red Hat Enterprise Linux 6 では、**SSH_USE_STRONG_RNG** および **OPENSSL_DISABLE_AES_NI** の値のみが ssh デーモン環境にエクスポートされました。変数 **AUTOCREATE_SERVER_KEYS** は、RSA と DSA サーバーのプライベートキーおよびパブリックキーを自動的に生成するのを init スクリプトに指示するために使用されました。

Red Hat Enterprise Linux 7 では、**export** コマンドを使用して、設定対象のサービス環境にこれらの値をエクスポートする必要はなくなりました。このため、以下の例では `/etc/sysconfig/sshd` ファイルが、これら 3 つの値をすべて ssh デーモンの環境にエクスポートします。

```
AUTOCREATE_SERVER_KEYS=YES
SSH_USE_STRONG_RNG=1
OPENSSL_DISABLE_AES_NI=1
```

2.4.3. 新ログインフレームワーク

Red Hat Enterprise Linux 7 では、**systemd** への移行の一環として、新しいログインデーモン (**journald**) が導入されました。**journald** は、全サービスに対して以下のタイプのメッセージをキャプチャーします。

- **syslog** メッセージ
- kernel メッセージ
- initial RAM ディスクおよび初期ブートメッセージ
- 標準出力および標準エラー出力に送信されるメッセージ

その後、これらのメッセージはネイティブのジャーナルファイルに保存されます。ジャーナルファイルは、構造化されインデックス化されたバイナリーファイルで、有用なメタデータを含み、容易かつスピーディーに検索ができます。

ジャーナルファイルは、デフォルトでは永続的に保存されません。ログに記録されるデータ量は、利用可能な空きメモリーの量によります。メモリーもしくは `/run/log/journal` ディレクトリーで容量が不足すると、一番古いジャーナルファイルが削除され、ログインを継続します。

Red Hat Enterprise Linux 7 では、**rsyslog** と **journald** が共存しています。**journald** が収集したデータは **rsyslog** に転送され、ここでさらなる処理が行われ、テキストベースのログファイルが保存されます。デフォルトでは、**rsyslog** は、**syslog** メッセージ用の標準的なジャーナルフィールドのみを保存しますが、**journald** で利用可能なすべてのフィールドを保存するように設定することもできます。つまり、Red Hat Enterprise Linux 7 は、依然として **rsyslog** に依存するアプリケーションおよびシステム設定と互換性を保っています。

ログインサブシステムの詳細については、[システム管理者ガイド](#)を参照してください。

2.4.4. ローカリゼーション設定

新 init システムである **systemd** への移行の一環として、ローカリゼーション設定は、`/etc/sysconfig/i18n` から `/etc/locale.conf` および `/etc/vconsole.conf` に移動しました。

2.4.5. ホスト名の定義

Red Hat Enterprise Linux 6 では、**hostname** 変数は **/etc/sysconfig/network** 設定ファイルで定義されていました。Red Hat Enterprise Linux 7 では、新 init システム (**systemd**) への移行時に、**hostname** 変数が **/etc/hostname** に定義されます。

2.4.6. Yum の更新

Red Hat Enterprise Linux 7 には更新バージョンの **yum** が含まれており、これには多くの変更および機能強化が含まれています。このセクションでは、**yum** を使用して Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 に移行する際に影響を受ける可能性のある変更を一覧表示しています。

- **yum group** および **yum groups** はトップレベルのコマンドとなり、コマンドラインの **yum** の使用における一貫性が改善されています。たとえば、**yum groupinfo** コマンドは、**yum group info** コマンドに変わりました。
- **yum group list** には、出力を変更するための新たなオプションのパラメーターが含まれています。この新たなオプションは、**language** と **ids** です。
- **/etc/yum.conf** 内の **group_command** パラメーターのデフォルト値が、**compat** から **objects** に変更になりました。これまで、**yum group install** のデフォルト動作は、パッケージグループの全メンバーをインストールし、以前にインストールされたパッケージと、以前のアップグレード以降にグループに追加されたパッケージの両方をアップグレードするというものでした。新たなデフォルト動作では、**yum** が、以前にインストールされたグループの記録を取り、グループの一部としてインストールされたパッケージと、別途インストールされたパッケージを区別します。
- **yum-security** プラグインおよび **yum-presto** プラグインは、**yum** に統合されました。
- **yum** は、複数のパッケージを同時にダウンロードできるようになっています。
- **yum** には環境グループのサポートが含まれています。これにより、環境グループ下にリストされている複数のパッケージグループを、単一エンティティとしてインストール、削除することができます。
- **yum** はリポジトリをパッケージのセットとして扱うことができるようになっていたため、リポジトリ内の全パッケージを単一エンティティとして扱うことが可能です。たとえば、そのリポジトリ内の全パッケージをインストールしたり、削除したりできます。この機能は、**repository-packages** サブコマンドが提供しています。
- **yum** に **--disableincludes** オプションが追加されました。これにより、設定ファイルで定義されている **include** ステートメントを無効にできるようになりました。**all** 値ですべての **include** ステートメントを無効にするか、特定のリポジトリ ID を提供することで、そのリポジトリに定義されている **include** ステートメントを無効にすることが可能です。
- **yum** に **--assumeno** オプションが追加されました。これは、**yum** による質問への回答が **no** であることを前提としています。このオプションは **--assumeyes** オプションを上書きしますが、**alwaysprompt** が規定する動作に依存します。

yum についての詳細情報は、**man** ページを参照してください。

```
$ man yum
```

2.4.7. RPM Package Manager (RPM) の更新

Red Hat Enterprise Linux 7 は、RPM Package Manager の更新バージョンを提供します。この更新には、移行に影響を与える可能性のある動作の変更が数多く含まれています。

- 競合検出はより厳密、正確になっています。競合の感度が高いため、Red Hat Enterprise Linux 6 にインストールされたパッケージがすべて Red Hat Enterprise Linux 7 にインストールされるとは限りません。
- 該当パッケージの他のバージョンと競合するパッケージを、代替方法を使ってシングルトンとして設定することができます。こうすることで、単一パッケージの複数のバージョンを同時にインストールすることが可能です。
- インストール済みのパッケージが別のパッケージを廃止予定としてリストしている場合、この2番目のパッケージはインストールされません。
- 廃止ルールには、アーキテクチャーなどの属性に関わらず、すべての適合パッケージが含まれます。
- 依存計算では、インストールされていないファイルや置き換えられたファイル(たとえば、**--nodocs**、**--noconfig**、または **--force** オプションなど)は考慮されません。
- パニックとなった (**DB_RUNRECOVER**) RPM Package Manager データベースを再構築する際に、**rm -f /var/lib/rpm/__db.** を手動で実行する必要がなくなりました。
- OpenPGP 3 で作成されたパブリックキーはサポートされません。
- **--info** オプションは、人間が読みやすくするために行ごとの個別のタグと値のペアを出力するようになりました。以前の **--info** 形式に依存するスクリプトは、書き換える必要があります。
- スペックパーサーはより厳密、正確になっており、これまでは受け入れられたスペックファイルでも、パーサーに失敗するか、警告が発せられる可能性があります。
- **%license** は、ライセンスとしてスペックファイルの **%files** セクションのファイルをマークするのに使用できるようになりました。このライセンスは、**--nodocs** が指定されたときでさえインストールする必要があります。
- バージョン比較は dpkg スタイルのチルダ (~) 演算子をサポートし、リリース前のソフトウェアを問題なく処理します。たとえば、**foo-2.0~beta1** は **foo-2.0** よりも古いとみなされるため、これら共通のアップストリームバージョンのプラクティスを処理するリリースフィールドに関する裏技の必要性がなくなります。
- 自動依存関係ジェネレーターは、ビルトインフィルター付きで、拡張可能かつカスタマイズ可能なルールベースのシステムに書き換えられました。

この更新には、以下の機能強化も含まれます。

- パッケージからインストールされたファイル (**INSTFILENAMES**)、ファイルへのハードリンクの数 (**FILENLINKS**)、パッケージバージョンコントロールシステムの詳細 (**VCS**)、およびフォーマット済みの依存関係文字列のショートカット (**PROVIDENEVRS**、**REQUIRENEVRS**、**CONFLICTNEVRS**、**OBSOLETENEVRS**) へのクエリーが可能になりました。
- 以下のものを含む多くのコマンドが新たに提供されました。
 - **rpmkeys**
 - **rpmdb**
 - **rpmspec**
 - **rpmsign**

- RPM Package Manager には、ランタイムマクロ拡張やランタイムクエリー形式の拡張を有効にするスクリプトレットへの新たなスイッチが含まれています。
- トランザクション前および後のスクリプトレット依存関係は、**Requires(pretrans)** および **Requires(posttrans)** で正確に表記されるようになっています。
- RPM Package Manager には **OrderWithRequires** タグが含まれており、ユーザーはこれで追加の順序付け情報を供給することができます。この新たなタグは Requires タグと同じ構文を使用しますが、依存関係は生成しません。同一トランザクション内に上記のパッケージが存在する場合は、トランザクションの順序付けを計算する際に、順序付けのヒントが **Requires** のように扱われます。
- スペックファイル内の行連結およびマクロ拡張が、指定された長さに制限されることはなくなりました。
- RPM Package Manager では、ユーザーがアップストリームバージョンコントロールのリポジトリ情報を指定できるようになりました。
- RPM Package Manager には、パッチ適用のプロセスの自動化を支援する **%autosetup** マクロが含まれています。

2.4.8. ifconfig の新フォーマット

廃止予定の **ifconfig** ツールからの出力形式が、Red Hat Enterprise Linux 7 で変更されています。**ifconfig** 出力を解析するスクリプトはこれらの変更に影響を受ける可能性があり、書き換えが必要な場合があります。

Red Hat では、廃止予定の **ifconfig** ツールではなく、**ip** ユーティリティとそのサブコマンド (**ip addr**、**ip link**) の使用を推奨しています。

2.4.9. コントロールグループの変更

カーネルは、システムリソースを管理する目的で、コントロールグループを使用してプロセスをグループ化します。Red Hat Enterprise Linux 7 では、コントロールグループに多くの変更が導入されています。

- コントロールグループは、**/cgroup** ではなく、**/sys/fs/cgroup** 下にマウントされています。
- ファイルシステムのなかには、デフォルトでマウントされているものもあります。
- **systemd** は、**libcgroup** から **systemd** への移行を完全にはサポートしていません。したがって、**cgred** サービスは、**systemd** が管理していないグループにプロセスを移動する場合にのみ使用してください。**cgconfig.conf** ファイルは、**systemd** が管理していないファイルシステムもしくはファイルコントローラーのコントロールグループ階層を設定する場合に使用してください。

これらの変更の詳細については、[リソース管理ガイド](#)を参照してください。

2.4.10. カーネルクラッシュ収集 (Kdump) の変更

これまで、カーネルクラッシュ収集ツールである **kdump** は、**kdump** キャプチャーカーネル用に、カスタム **mkdumprd** スクリプトで初期 RAMDisk (**initrd**) を生成していました。Red Hat Enterprise Linux 7 では、初期 RAMDisk が **dracut** で生成されるようになり、初期 RAMDisk 生成プロセスの維持が容易になっています。

この結果、**kdump** とその設定ファイルには以下の変更がなされました。

- **net** ディレクティブはサポート対象外となりました。 **ssh** もしくは **nfs** を明示的に定義する必要があります。
- **blacklist** オプションに対応しなくなりました。代わりに **rd.driver.blacklist** を、キャプチャーカーネルの **/etc/sysconfig/kdump** ファイル内のパラメーターとして指定することができます。
- デフォルトの **mount_root_run_init** アクションは、対象ターゲットへのダンプが失敗した場合に実行されていましたが、これは **dump_to_rootfs** アクションに置き換えられました。実際の root ファイルシステムをマウントして init スクリプトを実行し、**kdump** サービス開始時に vmcore の保存を試みる代わりに、root ファイルシステムをマウントして、即座に vmcore をそこに保存します。
- 新たなディレクティブである **dracut_args** は、kdump の設定時に追加の dracut 引数の指定を可能にします。
- **debug_mem_level** オプションが **kdump** に含まれなくなりました。この機能は dracut に移動しました。ユーザーは、キャプチャーカーネルの **/etc/sysconfig/kump** ファイルのパラメーターとして **rd.memdebug** を指定することで、同じ機能が実行できます。
- **options** ディレクティブはこれまで、初期 ram ファイルシステム (**initramfs**) 内のカーネルモジュール固有のパラメーターを含めるために使用されていました。この方法は、Red Hat Enterprise Linux 7 ではサポートされていません。代わりに、キャプチャーカーネルの **/etc/sysconfig/kdump** ファイルで関連パラメーターを指定できます。
- **link_delay** および **disk_timeout** パラメーターは不要となり、サポート対象外となりました。今まで、これらのパラメーターが必要とされていたユースケースに対処する **udev** が、dracut に含まれているためです。
- ファイルシステムのバックエンドダンプターゲットは、**kdump** サービスが開始し、初期 RAMDisk イメージが作成される前に、クラッシュしたカーネルにマウントする必要があります。そのターゲットを **/etc/fstab** に追加すると、システムの起動時に自動的にマウントされるようになります。
- パスを指定してもターゲットを指定せず、指定したパス内のディレクトリーが別のデバイスのマウントポイントであった場合、vmcore は、そのパスのどこかにマウントされたデバイスではなく、パスそのものに保存されます。このため、システムが再起動してデバイスがマウントされると、vmcore にはアクセスできなくなります。これは、デバイスがその位置の上にマウントするためです。Red Hat Enterprise Linux 7 は、ターゲットを指定せずにパスを指定した場合、この問題について警告します。

kdumpの詳細については、[カーネル管理ガイド](#)を参照してください。

2.4.11. usermod の動作の変更

Red Hat Enterprise Linux 6 では、**usermod** コマンドの **-g** オプションではグループ所有権を操作しませんでした。Red Hat Enterprise Linux 7.0 から Red Hat Enterprise Linux 7.2 リリースの **-g** オプションでは、**/home** ディレクトリーツリー内のファイルのグループ所有権が変更されていました。Red Hat Enterprise Linux 7.3 以降、**usermod** は、home ディレクトリーのユーザー ID が、修正したユーザー ID に一致した場合に限り、ユーザーのホームディレクトリーにあるファイルのグループ所有権を変更します。

2.4.12. システムアカウントへの変更

Red Hat Enterprise Linux 7 リリースでは、システムユーザー、および一般ユーザーおよびグループにおいて、デフォルトの ID 範囲が以下のように変更になりました。

表2.4 ID レイアウト

範囲	Red Hat Enterprise Linux 6	Red Hat Enterprise Linux 7
システムアカウント	0-499	0-999
ユーザーアカウント	500-60,000	1,000-60,000

この変更により、既存のユーザーの UID と GID に 500 ~ 999 を使用している場合に Red Hat Enterprise Linux 7 に移行すると、問題が発生する場合があります。UID および GID におけるデフォルトの範囲は、`/etc/login.defs` ファイルで手動で変更できます。

2.4.13. hwclock の動作の変更

Red Hat Enterprise Linux 6 では、ハードウェアクロックにアクセスするための `hwclock` コマンドが、システムのシャットダウンまたは再起動のたびに自動的に実行されていました。この動作は、Red Hat Enterprise Linux 7 の登場で変わります。システムクロックが **Network Time Protocol (NTP)** または **Precision Time Protocol (PTP)** で同期される場合は、カーネルが 11 分ごとに自動的にハードウェアクロックをシステムクロックに同期します。

NTP、**PTP** の設定、およびハードウェアクロックの設定の詳細については、[システム管理者ガイド](#)を参照してください。

2.5. ファイルシステムのフォーマット

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされたファイルシステムフォーマットのサポート変更の概要について説明しています。

2.5.1. 新デフォルトファイルシステム: XFS

XFS は非常に高パフォーマンスのスケラブルなファイルシステムで、日常的に要求が非常に高い環境で導入されています。Red Hat Enterprise Linux 7 では、XFS がデフォルトのファイルシステムとなっており、全アーキテクチャーでサポートされています。

Ext4 サイズ拡張は XFS のものと同じではありませんが、全アーキテクチャーで完全にサポートされており、引き続きアクティブな開発とサポートが行われます。

XFS に関する Red Hat サポート制限の詳細は、<https://access.redhat.com/site/articles/rhel-limits> から入手できます。

XFS ファイルシステムの使用と管理の詳細については、[ストレージ管理ガイド](#)を参照してください。

2.5.1.1. マウントオプションの変更

ext3 と ext4 とは異なり、XFS ファイルシステムはデフォルトで `user_xattr` および `acl` マウントオプションを有効にします。つまり、これらのオプションをコマンドラインもしくは `/etc/fstab` に含めると、以下のようなエラーが出ることになります。

```
$ mount -o acl /dev/loop0 test
mount: wrong fs type, bad option, bad superblock on /dev/loop0,
missing codepage or helper program, or other error
```

In some cases useful info is found in syslog - try
dmesg | tail or so.

ext3 および ext4 ファイルシステムは、デフォルトではその属性を有効にせず、**mount** コマンドもしくは **/etc/fstab** でこのファイルシステムをマウントする場合に、そのオプションを受け付けます。

2.5.2. Btrfs テクノロジープレビュー

Red Hat Enterprise Linux 7 では、テクノロジープレビューとして btrfs を導入しています。Btrfs は次世代 Linux ファイルシステムで、高度な管理、信頼性、および拡張性機能を提供します。Btrfs はメタデータとともにファイルのチェックサム検証を提供します。また、スナップショットおよび圧縮の機能、統合デバイス管理を提供します。

XFS に関する Red Hat サポート制限の詳細は、<https://access.redhat.com/site/articles/rhel-limits> から入手できます。テクノロジープレビュー機能に対するサポートレベルの詳細は、<https://access.redhat.com/site/support/offerings/techpreview/> を参照してください。

btrfs の使用と管理の詳細については、[ストレージ管理ガイド](#)を参照してください。

2.5.2.1. btrfs のキックスタート

キックスタートファイルでは、システムでパーティションを作成する場合に、通常は以下のように **part** コマンドを **--fstype** とともに使用して、特定のファイルシステムを使用したパーティションを作成します。

```
part /mnt/example --fstype=xfs
```

ただし、Red Hat Enterprise Linux 7.0 および 7.1 では、btrfs はファイルシステムタイプではなくデバイスタイプとして取り扱われます。したがって、**btrfs** は **--fstype** パラメーターの有効な値ではありません。代わりに、以下のように **btrfs** コマンドを使用して btrfs ボリュームを定義します。

```
btrfs mount_point --data=level --metadata=level --label=label partitions
```

2.5.3. 拡張ファイルシステムのサポート

Red Hat Enterprise Linux 7 では、Ext2、Ext3、および Ext4 のサポートを提供する統一拡張ファイルシステムドライバーが導入されています。

ただし、Ext2 は Red Hat Enterprise Linux 7 では廃止予定とみなされており、可能であれば使用を避けてください。

これらのファイルシステムの詳細については、[ストレージ管理ガイド](#)を参照してください。

2.6. 物理ストレージ

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされた物理ストレージと関連設定ツールのサポート変更の概要について説明しています。

2.6.1. 起動時のマウント動作の変更

ストレージデバイスが起動時にマウントされるよう設定されているにも関わらず、そのデバイスを見つけることができない、またはそのデバイスが正常にマウントされないと、Red Hat Enterprise Linux 7 は

起動に失敗します。これは、重要なストレージデバイスがなくてもシステムが起動することを防ぐために意図的に変更した動作です。Red Hat Enterprise Linux の以前のバージョンは、起動時にマウントされるように設定されたすべてのストレージデバイスが検出されたか、もしくは正しくマウントされたかに関係なく起動されました。

デバイスのシステムが起動しないことがないようにする必要がある場合は、以下に示されたように **nofail** オプションでデバイスをマークできます。

```
/dev/essential-disk /essential xfs auto,defaults 0 0
/dev/non-essential-disk /non-essential xfs auto,defaults,nofail 0 0
```

2.6.2. LVM スナップショットをロールバックメカニズムとして使用



警告

LVM スナップショットは、第1のロールバック方法としては推奨されません。アップグレード中にはシステム全体 (ユーザーファイルを除く) が上書きされます。このため、システムのスナップショットは元のデータセットとほとんど同じサイズになります。

さらに、スナップショットには **/boot** パーティションが含まれないことから、通常のバックアッププロセスよりもエラーが発生しやすくなります。

Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 へのアップグレードの際には、完全なバックアップをとり、これを第1のロールバック方法として使用することを Red Hat では推奨しています。LVM スナップショットは、2 番目のロールバック方法としてのみ使用してください。

Red Hat Enterprise Linux 6.3 では、論理ボリューム上で残っているスペースをスナップショットのストレージスペースとして使用することができます。アップグレードや移行が失敗した場合は、システムはこのスナップショットに戻すことができます。

LVM スナップショットを第2のロールバック方法として使用したい場合は、完全なスナップショットに余裕を持たせるため、スペースを追加する必要があるかもしれません。スペースの追加は、以下のいずれかで実行できます。

- 新たなディスクを追加します。手順については、[ストレージ管理ガイド](#)を参照してください。
- **parted** を使って、既存のパーティションに割り当てられていない空きスペースをチェックします。
- **lsblk** を使って、空きパーティションもしくは削除して空きスペースにできるパーティションをチェックします。
- **vgdisplay** を使って、論理ボリュームに割り当てられていないボリュームグループに空き領域があるかを確認します。
- **df** を使って、空き領域があり縮小可能なファイルシステムを確認します。この場合、論理ボリュームもしくはパーティションを縮小して空き領域にすることができます。

ロールバックに LVM スナップショットを使用する際には、以下の制限がある場合があることに注意してください。

- スナップショットのサイズは自動的に調整されません。パーティションに対してスナップショットが大きくなり過ぎた場合は、スナップショットが無効になる可能性があり、その場合はロールバックが失敗することになります。このため、スナップショットを作成する前に、システム全体のスナップショットに十分な大きさの領域を割り当てることが必須となります。root スナップショットのサイズ変更が必要な場合は、元の root デバイスをアンマウントしてサイズを変更している間に root デバイスとして使用可能な、ライブ CD などの新たなデバイスが必要になります。
- スナップショットのコピーオンライトのデバイスは複製されず、システムが複製されるかどうかに関わらず、単一デバイス上に置かれます。このデバイスが失敗してスナップショットが失われると、ロールバックができなくなります。Red Hat では、mdraid による物理ボリュームの使用、もしくは複数のスナップショットを使ったディスクの分割を推奨しています。複数のスナップショットを使用したほうが遅くなります。
- インストール中にクラッシュが発生すると、システムは起動できなくなります。このような場合、Red Hat では、ライブ CD もしくは PXE ブートで起動し、システムが正常に起動した後にスナップショットをマージすることを推奨しています。マージの手順は、[Red Hat Enterprise Linux 7 LVM のドキュメント](#)に記載されています。
- ロールバックすると、`/var/log` をアップグレード前の状態に戻します。監査のために、Red Hat は、ロールバック開始前に、ログファイルをインストールから別の場所にコピーすることを推奨します。

2.6.3. targetcli によるターゲット管理

Red Hat Enterprise Linux の以前のバージョンでは、iSCSI ターゲットのサポートには `tgt` を使用し、Linux カーネルターゲットの LIO は `fcoe-target-utils` パッケージで Ethernet 経由のファイバーチャネル (FCoE) ターゲットにのみ使われていました。

Red Hat Enterprise Linux 7 では、LIO カーネルターゲットサブシステムを FCoE、iSCSI、iSER (Mellanox InfiniBand)、および SRP (Mellanox InfiniBand) ストレージファブリックに使用します。ファブリックはすべて、`targetcli` ツールで管理できます。

2.6.4. 永続的なデバイス名

Red Hat Enterprise Linux 7 では、カーネルメッセージ内にデバイス名 (たとえば、`sda`、`sdb` など) と永続的なデバイス名 (`udev` が `/dev/disk/by-*` で提供) のマッピングを保存することで、システム上のデバイス管理を容易にしています。これにより、システム管理者は、ブートするたびにデバイス名が変更しても、特定のデバイスに関連付けられたメッセージを特定することができます。

カーネルの `/dev/kmsg` ログは `dmesg` コマンドで表示でき、シンボリックリンク用のメッセージを表示できます。これは、`udev` がカーネルデバイス用に作成したものです。これらのメッセージは、以下の形式で表示されます。`udev-alias: device_name (symbolic_link symbolic link ...)`以下に例を示します。

```
udev-alias: sdb (disk/by-id/ata-QEMU_HARDDISK_QM00001)
```

このメッセージは、どのログアナライザーでも表示できます。また、メッセージは `syslog` により `/var/log/messages` に保存されます。

この機能を有効にするには、`/etc/default/grub` のカーネルコマンドラインに `udev.alias=1` を追加します。

2.6.5. LVM キャッシュボリューム

Red Hat Enterprise Linux 7.1 では、LVM キャッシュボリューム機能が完全に対応するようになりました。この機能を使用すると、小規模で高速なデバイスで論理ボリュームを作成し、大規模で低速なデバイスのキャッシュとして動作させることができますようになります。キャッシュ論理ボリュームの作成方法については **lvmpcache** の man ページを参照してください。

2.7. ネットワーク

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされたネットワークング、ネットワークプロトコルサポート、および関連設定ツールの変更の概要について説明しています。

2.7.1. 推奨される命名プラクティス

ホスト名は、最大 64 文字の長さで自由形式の文字列になります。ただし、Red Hat では、static および transient の両方の名前が **host.example.com** のように DNS 内のマシンで使われている完全修飾ドメイン名 (FQDN) に合致することを推奨しています。hostnamectl ツールを使うと、a-z、A-Z、0-9、-、. のみを使用して最大 64 文字の長さの静的および一時的なホスト名が可能になります。現在の仕様では、アンダースコアが技術的に許容されます。ただし、以前の仕様で禁止されているため、Red Hat では、ホスト名にアンダースコアを使用することは推奨していません。

ICANN (The Internet Corporation for Assigned Names and Numbers) は、(.yourcompany などの) トップレベルの未登録ドメインを公開登録簿に追加することがあります。このため、Red Hat では、プライベートネットワーク上であっても委任されていないドメイン名を使用しないことを強く推奨しています。これは、ネットワーク設定によっては異なる解決をしてしまうドメインネームになってしまう可能性があるからです。その結果、ネットワークリソースは利用できなくなります。また、委任されていないドメイン名を使うと、DNSSEC の実装および維持がより困難になります。これは、ドメイン名の競合が DNSSEC 検証に手動の設定ペナルティーを加えることになるからです。

この問題の詳細については、[ドメイン名の競合に関する ICANN FAQ](#) を参照してください。

2.7.2. NetworkManager の更新

Red Hat Enterprise Linux 7 には、NetworkManager の更新バージョンが含まれており、多くの機能強化と新機能をいくつか提供しています。

- nmcli ツールは、**nmcli con edit** および **nmcli con modify** コマンドを使った接続の編集をサポートします。
- ネットワーク設定およびネットワーク接続管理には、新たなテキストベースのユーザーインターフェイス (nmtui) が簡素化されたコンソールベースのツールを提供します。これは、system-config-network-tui ツールに代わるものです。
- これまで NetworkManager は、認識しないインターフェイス (Ethernet、Infiniband、WiFi、Bridge、Bond、および VLAN 以外のインターフェイス) を無視していました。現在は、ip link が検出したネットワークインターフェイスはすべて NetworkManager が認識するようになり、これを nmcli のような D-Bus インターフェイスやクライアントで公開します。これにより、NetworkManager は ip のようなツールと同様のものになっています。
- NetworkManager は、Ethernet、InfiniBand、Bridge、Bond、VLAN、Team など、ネイティブに設定可能なインターフェイスを非破壊的に所有するようになりました。これらのインターフェイスが NetworkManager の起動もしくは再起動前に設定されても、以前に設定された接続は切断されません。これは、NM_CONTROLLED オプションが必要なくなったことを意味します。

- ネットワーク接続性、ホットスポット、ポータルのチェックをサポートします。この動作は、デフォルトでは無効になっています。
- チームインターフェイスをサポートします。
- GRE、macvlan、macvtap、tun、tap、veth、および vxlan デバイスの基本的かつ非ネイティブサポートです。
- 新たな **NetworkManager-config-server** パッケージが、サーバーに適切なデフォルトを提供します。たとえば、キャリアの変更無視や、デフォルト DHCP 接続を作成しないなどです。
- **NetworkManager.conf** の新たな **dns=none** 設定オプションにより、**NetworkManager** が **resolv.conf** ファイルを変更しないようにします。
- ユーザーのすばやいスイッチングをサポートします。
- インターフェイスの MAC アドレスに追加、もしくはその代わりに使用するインターフェイス名への接続固定をサポートします。

この更新により、動作をモニターする設定ファイルも変更されます。**NetworkManager** は、ディスク上の設定ファイルの変更をモニターしなくなりました。代わりに、**nmcli con reload** コマンドで手動で変更した設定ファイルをリロードする必要があります。

2.7.3. 新ネットワーク命名スキーマ

Red Hat Enterprise Linux 7 は、ネットワークインターフェイス用に一貫した予想可能なネットワークデバイス命名の方法を提供します。この機能では、インターフェイスの位置判定と区別が容易になるようにシステム上のネットワークインターフェイス名を変更します。

従来、Linux のネットワークインターフェイスは **eth[0123...]** として列挙されていましたが、これらの名前は必ずしもシャーシの実際のラベルに対応しているとは限りません。複数のネットワークアダプターを使用する最新のサーバープラットフォームでは、このインターフェイスの非決定論的および反直感的な命名が行われています。これは、マザーボードに組み込まれたネットワークアダプター (Lan-on-Motherboard、もしくは LOM) とアドイン (シングルおよびマルチのポート) アダプターの両方に影響します。

Red Hat Enterprise Linux 7 では、**systemd** および **udev** が多くの異なる命名スキームをサポートしています。デフォルトの動作では、ファームウェア、トポロジー、および場所情報に基づいて固定名が割り当てられます。これは、名前が完全に自動的かつ予想可能であり、ハードウェアが追加もしくは削除されても (再列挙がなされず) 固定のままであり、またハードウェアが壊れた場合にシームレスに交換できるという利点があります。マイナス面は、従来使用されていた名前と比べて読みにくい場合があるという点です。たとえば、**eth0** であったものが **enp5s0** になるというようにです。

以下のネットワークインターフェイス用の命名スキームは、**udev** がネイティブにサポートしています。

スキーム 1

eno1 など、ファームウェアや BIOS が提供するオンボードデバイスのインデックス番号を含めて命名します。デフォルトでは、ファームウェアからの情報が適用可能かつ利用可能である場合に、**systemd** は、このスキームに基づきインターフェイスの名前を付け、スキーム 2 をフォールバックとして使用します。

スキーム 2

ens1 など、ファームウェアや BIOS が提供する PCI Express ホットプラグスロットのインデックス番号が含まれて命名します。デフォルトでは、ファームウェアからの情報が適用可能かつ利用可能である場合に、**systemd** は、このスキームに基づきインターフェイスの名前を付け、スキーム 3 を

フォールバックとして使用します。

スキーム 3

enp2s0 など、ハードウェアのコネクタの物理的な場所を含めて命名します。デフォルトでは、ファームウェアからの情報が適用可能かつ利用可能である場合に、**systemd** は、このスキームに基づきインターフェイスの名前を付け、スキーム 5 をフォールバックとして使用します。

スキーム 4

組み入れるインターフェイスの MAC アドレスを命名します。たとえば、**enx78e7d1ea46da** です。デフォルトでは、**systemd** はインターフェイスをこのスキームにしたがって命名しませんが、必要に応じて有効にすることができます。

スキーム 5

eth0 など、従来の予測できないカーネルネイティブの **ethX** で命名します。**systemd** は、他のすべての方法が失敗した場合に、このスキームを基にインターフェイスを命名します。

システムで **BIOSDEVNAME** を有効にしている場合、もしくはユーザーがカーネルデバイスの名前を変更する **udev** ルールを追加している場合は、これらのルールがデフォルトの **systemd** ポリシーに優先されます。

この新しい命名システムの詳細については、[ネットワークガイド](#)を参照してください。

2.7.4. 新しいネットワーキングユーティリティー (ncat)

Red Hat Enterprise Linux 7 の **netcat** の代替として、新たに追加されたネットワークユーティリティー **ncat**。 **ncat** は、信頼できるバックエンドツールで、他のアプリケーションやユーザーとネットワーク接続できるようにします。コマンドラインからデータの読み取りと書き込みを行い、通信に TCP と UDP の両方を使用します。

ncat のコマンドのいくつかは、**netcat** が元々提供していたものとは異なるか、同じオプションでも異なる機能を提供します。この違いは、以下のリストで要約されています。

- **netcat -P** オプションは、認証が必要なプロキシサーバーに提示するユーザー名をとっていました。この動作を行う **ncat** オプションは、**--proxy-auth user[:pass]** になります。
- **netcat -X** オプションは、プロキシサーバーとの通信時に使用するネットワークユーティリティー用に指定されたプロトコルをとっていました。この動作を行う **ncat** オプションは **--proxy-type** です。
- **netcat -x** オプションは、プロキシサーバーと接続するためのネットワークユーティリティー用のアドレスおよびオプションのポートをとっていました。この動作の **ncat** オプションは **--proxy** です。これは IP アドレスとオプションのポート (例: **--proxy host[:port]**) を取ります。
- **netcat -d** オプションは、stdin からの読み取りを無効にしていました。**ncat -d** オプションでは、ユーザーが読み取りと書き込み操作間の待ち時間を指定することができます。ただし、**ncat** では、**netcat -d** と同様に動作する **---recv-only** オプションを利用できます。
- **netcat -i** オプションは、テキスト行の送受信間隔または複数ポートへの接続間隔を指定していました。**ncat -i** オプションでは、接続がタイムアウトして切断されるまでの待機時間を指定します。**ncat** には **netcat -i** オプションと同様のものはありません。
- **netcat -w** オプションは、確立できない接続がタイムアウトして切断されるまでの待機時間を指定していました。**ncat -w** オプションでは、タイムアウトまでの接続試行時間を指定します。

netcat で利用できたオプションによっては、**ncat** に同等のものがないものもあります。**Ncat** は現在、以下を実行できません。

- ソケット上でのデバッグの有効化 (以前は **netcat -D** が提供)。
- TCP 送受信バッファサイズの指定 (以前は **netcat -l** および **netcat -O** が提供)。
- 送信元もしくは宛先ポートがランダムに選択されることを指定 (以前は **netcat -r** が提供)。
- TCP MD5 シグネチャーオプション、RFC 2385 経由での BGP セッション保護の有効化 (以前は **netcat -S** が提供)。
- サービスの IPv4 タイプを指定 (以前は **netcat -T** が提供)。
- UNIX ドメインソケットの使用を指定 (以前は **netcat -U** が提供)。
- 使用するルーティングテーブルを指定 (以前は **netcat -V** が提供)。
- データの送信なしにリスニングデーモンをスキャン。
- テキスト行の送受信間隔または複数ポートへの接続間隔を指定。

ncat コーティリティーは **nmap-ncat** パッケージが提供します。**ncat** についての詳細情報は、**man** ページを参照してください。

```
$ man ncat
```

2.7.5. Postfix の変更点

Red Hat Enterprise Linux 7 では、**postfix** はバージョン 2.6 からバージョン 2.10 にアップグレードされます。Red Hat Enterprise Linux 6 から 7 にアップグレードするときに、主要な互換性の問題は Preupgrade Assistant によって処理されますが、ユーザーは以下の致命的でない互換性の問題に注意する必要があります。

- **pass** マスターサービスでの問題を回避するために、**postscreen** デーモンを使用する前に **postfix stop** コマンドと **postfix start** コマンドを実行する必要があります。
- システムにより提供されるデフォルトの CA 証明書は、*_tls_CAfile または *_tls_CApath リストに追加されなくなりました。つまり、**permit_tls_all_clientcerts** が使用された場合、サードパーティー製の証明書はメールリレーパーミッションを受け取りません。設定に証明書の検証が必要である場合は、**tls_append_default_CA = yes** を設定することにより、後方互換性の動作を有効にします。
- **verify** サービスは、デフォルトで定期クリーンアップが有効な状態で永続キャッシュを使用するようになりました。削除操作とシーケンス操作のサポートが必要です。このキャッシュを無効にするには、**address_verify_map** で空の **main.cf** パラメーターを指定します。定期クリーンアップを無効にするには、**address_verify_cache_cleanup_interval** を **0** に設定します。
- 以前は、フィルターの次ホップ宛先が指定されていない場合に、デフォルトの次ホップ宛先が **\$myhostname** の値でした。このデフォルト値は受信者ドメインになりました。デフォルトの次ホップ宛先を変更するには、**default_filter_nexthop = \$myhostname** を指定します。パイプベースのフィルターでは、これにより、ラウンドロビンドメインが選択されずに FIFO 配信順序も有効になります。
- **postmulti -e destroy** コマンドを実行した場合に、**postmulti -e create** コマンドの実行後に作成されたファイルの削除が試行されなくなりました。
- Postfix は、Milter の **smfi_addrcpt** アクションで受信者を追加したときにデフォルトの配信ステータス通知を要求するようになりました。

- 仮想エイリアスの拡張の結果が仮想エイリアスの再帰または拡張の制限を超過したときに、Postfix は、余分な受信者を警告なしで破棄し、メッセージを配信する代わりに一時配信エラーを報告するようになりました。
- ローカル配信エージェントが、owner-alias を持たない子エイリアスにメールを配信するときに親エイリアスの owner-alias 属性を保持するようになりました。これにより、メーリングリストへの重複配信の可能性が少なくなります。古い動作を有効にするには、**reset_owner_alias = yes** を指定します。
- Postfix SMTP クライアントは、"." なしで DNS 名をルックアップするときにローカルドメインを追加しなくなりました。古い動作を有効にするには、**smtp_dns_resolver_options = res_defnames** を指定します。これにより、予期しない結果がもたらされることがあることに注意してください。
- **postfix/smtpd[pid]: queueid: client=host[addr]** ログファイルレコードの形式が変更されました。可能な場合は、レコードの最後に before-filter クライアント情報と before-filter キュー ID が追加されるようになりました。
- デフォルトでは、postfix は受信者が指定されていないメッセージに未公開の受信者ヘッダーを追加するようになりました。古い動作を有効にするには、**mail.cf** で以下の内容を指定します。

```
undisclosed_recipients_header = To: undisclosed-recipients;;
```

- SASL メカニズムリストは、**STARTTLS** が正常に完了したあとに常に再計算されるようになりました。
- **smtpd_starttls_timeout** のデフォルト値は、ストレス依存になりました。
- ドメイン名にシークレット部分がある DNSBL クエリーで、**postscreen** SMTP 返信からそのシークレット部分を隠すことが必要になりました。たとえば、**main.cf** で、以下のように指定します。

```
postscreen_dnsbl_reply_map = texthash:/etc/postfix/dnsbl_reply
```

dnsbl_reply で、以下のように個別の DNSBL 名を指定します。

```
# Secret DNSBL name  Name in postscreen(8) replies
secret.zen.spamhaus.org  zen.spamhaus.org
```

- VSTREAM エラーが読み取りおよび書き込みエラーに別のフラグを使用するようになったため、postfix VSTREAMs を使用するすべてのプログラムは再コンパイルする必要があります。
- **smtp_line_length_limit** のデフォルト値は、SMTP の標準値に合わせて **999** になりました。
- Sendmail は、**<CR><LF>** で終わるすべての入力行を UNIX 形式 (**<LF>**) に変換するようになりました。
- デフォルトでは、SMTP クライアントは **AUTH=<>** を **MAIL FROM** コマンドに追加しなくなりました。
- **fatal** と以前に分類された一部のログメッセージは、**error** と分類されるようになりました。それに応じて、ログファイルベースのアラートシステムを更新する必要がある場合があります。古い動作を有効にするには、**daemon_table_open_error_is_fatal** を **yes** に設定します。
- 新しくサポートされた長いキューファイル名は、Postfix 2.9 より前のバージョンではサポートされていません。Postfix 2.8 またはそれ以前のバージョンに移行するには、長いキューファイ

ル名のすべてを変換する必要があります。この場合は、postfix を停止し、**enable_long_queue_ids** を **no** に設定して、キューファイル名の変更をエクスポートしなくなるまで **postsuper** コマンドを実行します。

- Postfix は、TLS ロギングレベルが 0 の場合に正常な TLS ネゴシエーションの結果をログに記録するようになりました。ログレベルの詳細については、**postconf** の man ページを参照してください。
- postfix SMTP サーバーは、常に **smtpd_sender_login_maps** テーブルをチェックします。
- デフォルトの **inet_protocols** 値は **all** (IPv4 と IPv6 の両方を使用) になりました。グローバルな IPv6 接続がないサイトで予期しないパフォーマンスの損失を回避するために、**make upgrade** コマンドと **postfix upgrade-configuration** コマンドは、明示的な設定が存在しない場合に **inet_protocols = ipv4** を **main.cf** に追加します。
- デフォルトの **smtp_address_preference** 値は、**any** (IPv4 または IPv6 をランダムに選択) になりました。
- SMTP サーバーは、ルックアップテーブルが利用可能でないためクライアントのコマンドが拒否されたセッションの内容を報告しなくなりました。このようなレポートを引き続き受け取るには、**data** クラスを **notify_classes** パラメーターの値に追加します。
- 新しい **smtpd_relay_restrictions** パラメーターが追加されました。デフォルトでは、これにより **permit_mynetworks**、**permit_sasl_authenticated**、および **defer_unauth_destination** が有効になります。また、**smtpd_recipient_restrictions** のスパムフィルタールールの間違いにより発生するオープンリレーの問題が回避されます。ただし、**smtpd_recipient_restrictions** 下で複雑なメールリレーポリシーが設定されているサイトの場合は、一部のメールが間違えて遅延されることがあります。この問題を修正するには、**smtpd_relay_restrictions** 設定を削除するか、**smtpd_recipient_restrictions** の既存のポリシーを使用するか、既存のポリシーを **smtpd_recipient_restrictions** から **smtpd_relay_restrictions** にコピーします。

2.7.6. ネットワークプロトコル

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされたネットワークプロトコル変更の概要について説明しています。

2.7.6.1. Network File System (NFS)

Red Hat Enterprise Linux 7 は NFS 3、NFS 4.0 および NFS 4.1 をサポートしています。NFS 2 は Red Hat Enterprise Linux 7 ではサポート対象外となっています。

NFS 4.1 は、Parallel NFS (pNFS) のクライアントサポートを含む多くのパフォーマンスおよびセキュリティ機能強化を提供します。さらに、コールバックには別個の TCP 接続が不要となり、たとえば NAT やファイアウォールが妨害するなど NFS サーバーがクライアントにコンタクトできない場合でも NFS サーバーは委任を許可することができます。

NFS 3、NFS 4.0、NFS 4.1 がサーバー上でサポートされています。特定バージョンのサポートは、**/etc/sysconfig/nfs** ファイルで **RPCNFSDARGS** パラメーターを変更することで有効もしくは無効にできます。たとえば、**RPCNFSDARGS="-N4.1 -V3"** は NFS 3 のサポートを有効にし、NFS 4.1 のサポートを無効にします。詳細は、以下の man ページを参照してください。

```
$ man rpc.nfsd
```

NFS クライアントはデフォルトで NFS 4.0 を使ってマウントを試行し、マウント操作が失敗すると NFS 3 にフォールバックします。デフォルトの動作は **/etc/nfsmount.conf** ファイルを編集し、コマンドラインオプションを使用することで変更できます。詳細情報は、man ページを参照してください。

```
$ man nfs
```

```
$ man nfsmount.conf
```

2.7.6.1.1. Parallel NFS (pNFS)

Red Hat Enterprise Linux 7 では、Parallel NFS (pNFS) にクライアントサポートを提供しています。pNFS は、NFS のスケーラビリティを向上できます。また、パフォーマンスを向上する可能性があります。Red Hat Enterprise Linux 7 クライアントが pNFS に対応するサーバーをマウントすると、そのクライアントは、複数のサーバーから同時にデータにアクセスできるようになります。Red Hat Enterprise Linux 7 は、ファイルのレイアウトタイプをサポートします。また、オブジェクトとブロックのレイアウトタイプがテクノロジープレビューと同梱されています。このプロトコルとその機能の詳細については、[ストレージ管理ガイド](#)を参照してください。

2.7.6.2. Apache Web Server (httpd)

Red Hat Enterprise Linux 7 では、Apache Web Server の更新バージョンが提供されています。この新バージョン (2.4) には、多くの新機能に加えて重要なパッケージ変更も含まれています。

変更されたプロキシ設定

SSL バックエンドを使用する Apache Web Server (**httpd**) 設定では、設定されたホスト名に SSL 証明書が一致しない場合に **SSLProxyCheckPeerName** ディレクティブを使用する必要があります。以前は、プロキシバックエンドの SSL 証明書に記載されているホスト名が検証されました。

新制御メカニズム

Red Hat Enterprise Linux はシステムを SysV init スクリプトから移動しているため、**httpd** サービスを制御するコマンドが変更されました。Red Hat では、**service** コマンドではなく、**apachectl** および **systemctl** コマンドを推奨しています。たとえば、以前は **service httpd graceful** を実行していた場面では、**apachectl graceful** を実行することを Red Hat では推奨します。

デフォルトのサブコマンド動作の変更

httpd 用の **systemd** ユニットファイルが **reload** および **stop** サブコマンドの動作を定義します。具体的には、デフォルトで **reload** サブコマンドは正常にサービスをリロードし、**stop** コマンドはサービスを停止します。

ハードコーディングされたデフォルト設定

以前のバージョンの **httpd** は、すべての設定設定とデフォルトを一覧表示した網羅的な設定ファイルを提供していました。多くの共通設定設定は、デフォルト設定ファイルで明示的に設定されることはなくなりました。代わりに、デフォルト設定はハードコーディングされています。デフォルト設定ファイルにあるのは最小限のコンテンツで、その結果、管理が容易になっています。ハードコーディングされた全設定向けのデフォルト値はマニュアルで指定されており、これはデフォルトでは **/usr/share/httpd** にインストールされています。

新 Multi-Processing Model モジュール

Red Hat Enterprise Linux の以前のリリースでは、いくつかの Multi-Processing Models (**prefork** および **worker**) を異なる **httpd** バイナリーとして提供していました。Red Hat Enterprise Linux 7 では単一のバイナリーを使用し、これらの Multi-Processing Models を読み込み可能なモジュール **worker**、**prefork** (デフォルト)、および **event** として提供しています。読み込むモジュールを選択するには、**/etc/httpd/conf.modules.d/00-mpm.conf** ファイルを編集してください。

ディレクトリーの変更

この更新バージョンの **httpd** では、多くのディレクトリーが移動してしまったか、提供されていません。

- これまで `/var/cache/mod_proxy` にインストールされていたコンテンツは `/var/cache/httpd` に移動し、`proxy` または `ssl` サブディレクトリ下にあります。
- これまで `/var/www` にインストールされていたコンテンツは `/usr/share/httpd` に移動しました。
- これまで `/var/www/icons` にインストールされていたコンテンツは `/usr/share/httpd/icons` に移動しました。このディレクトリには、ディレクトリインデックスで使われるアイコンが含まれています。
- `httpd` マニュアルの HTML バージョンはこれまで `/var/www/manual` にインストールされていましたが、`/usr/share/httpd/manual` に移動しました。
- カスタムの多言語 HTTP エラーページはこれまで `/var/www/error` にインストールされていましたが、`/usr/share/httpd/error` に移動しました。

suexec の変更

`suexec` バイナリーには、インストール時に `root` に設定されていたユーザー ID がなくなりました。代わりに、より限定的なパーミッションセットがファイルシステムの機能を使って適用されます。これにより、`httpd` サービスのセキュリティが改善されます。また、`suexec` は `/var/log/httpd/suexec.log` を使用する代わりにログメッセージを `syslog` に送信します。`syslog` に送信されたメッセージは、デフォルトで `/var/log/secure` に現れます。

モジュールインターフェイス互換性の変更

`httpd` モジュールインターフェイスが変更したことで、この更新バージョンの `httpd` は、以前のバージョンの `httpd` (2.2) に構築されたサードパーティーのバイナリーモジュールとは互換性がないこととなります。これらのモジュールは、必要に応じて `httpd 2.4` モジュールインターフェイス用に調整し、再構築する必要があります。バージョン 2.4 における API 変更の詳細は、Apache ドキュメンテーションを参照してください。

apxs バイナリーの場所の変更

ソースからのモジュール構築に使用される `apxs` バイナリーは、`/usr/sbin/apxs` から `/usr/bin/apxs` に移動しました。

新設定ファイルおよび移動された設定ファイル

モジュールを読み込む設定ファイルは、`/etc/httpd/conf.modules.d` ディレクトリ内にあります。(php パッケージのような) `httpd` 用の追加の読み込み可能なモジュールを提供するパッケージは、ファイルをこのディレクトリに追加します。`conf.modules.d` ディレクトリ内の設定ファイルはすべて、`httpd.conf` の本文の前に処理されます。`/etc/httpd/conf.d` ディレクトリ内の設定ファイルは、`httpd.conf` の本文の後で処理されるようになりました。

`httpd` パッケージは追加の設定ファイルを提供しています。

- `/etc/httpd/conf.d/autoindex.conf` は、`mod_autoindex` ディレクトリーのインデックス作成を設定します。
- `/etc/httpd/conf.d/userdir.conf` は、ユーザーディレクトリー (<http://example.com/~username/>) へのアクセスを設定します。デフォルトでは、このアクセスはセキュリティのために無効になっています。
- `/etc/httpd/conf.d/welcome.conf` は、コンテンツがない場合に <http://localhost/> に表示されるようこのページを設定します。

設定互換性の変更

このバージョンの `httpd` は、以前のバージョン (2.2) の設定構文と互換性がありません。設定ファイルは、この更新バージョンの `httpd` で使用可能となる前に、構文を更新する必要があります。バージョン 2.2 から 2.4 で変更された構文の詳細については、Apache ドキュメンテーションを参照してください。

2.7.6.3. Samba

Red Hat Enterprise Linux 7 は Samba 4 を提供します。これは、デーモンとクライアントユーティリティー、SMB1、SMB2、SMB3 のプロトコルを使用した通信を可能にする Python バインディングを組み合わせたものです。

現行の Kerberos 実装は、Samba 4 Active Directory ドメインコントローラーの機能をサポートしていません。この機能は Red Hat Enterprise Linux 7.0 では省略されていますが、今後のリリースで導入される予定です。Active Directory DC に依存しないその他の機能はすべて、含まれています。

Red Hat Enterprise Linux 6.4 およびそれ以降では、Samba 4 はテクノロジープレビューとして提供され、**安定性のある Samba 3 パッケージ ([package]*samba-)** との競合を避けるために **[package]samba4-* パッケージシリーズとしてパッケージ化されていました**。Samba 4 は今回、完全にサポートされ、Samba 3 に関して多くの機能強化を提供しているため、Red Hat Enterprise Linux 7 では Samba 4 を標準 **[package]*samba- パッケージとして提供しています**。特殊な **[package]*samba4-** パッケージは廃止されました。

Samba の詳細は、[システム管理者ガイド](#)を参照してください。

2.7.6.4. BIND

Red Hat Enterprise Linux 6 では、`bind-chroot` パッケージをインストールすると、`/etc/sysconfig/named` の `chroot` 環境の場所を参照する `ROOTDIR` 環境変数が変更されました。`named` サービスを (`chroot` 環境ではなく) 正常に実行するには、`bind-chroot` パッケージを削除するか、`/etc/sysconfig/named` ファイルで `ROOTDIR` 環境変数を手動で編集する必要がありました。

Red Hat Enterprise Linux 7 では、`bind-chroot` パッケージをインストールしても、`named` サービスが実行される方法は変わりません。代わりに、新しいサービスである `named-chroot` がインストールされます。このサービスは、以下のように `systemctl` コマンドを使用して起動または停止できます。

```
# systemctl start named-chroot.service
```

```
# systemctl stop named-chroot.service
```

`named-chroot` サービスは、`named` サービスと同時に実行できません。

2.7.7. デフォルトの製品証明書

Red Hat Enterprise Linux 7.2 リリース以降では、デフォルトの証明書が `redhat-release` パッケージに追加されています。このデフォルトの証明書は、`/etc/pki/product-default/` ディレクトリーに保存されています。

サブスクリプションマネージャーは、`/etc/pki/product/` ディレクトリーの証明書一覧を調べてから、`/etc/pki/product-default/` ディレクトリーを調べます。`/etc/pki/product-default/` ディレクトリーのコンテンツは `redhat-release` パッケージが提供します。`/etc/pki/product/` に置かれていない `/etc/pki/product-default/` ディレクトリーの証明書は、インストールされているとみなされます。サブスクリプションマネージャーが、サブスクリプションしているチャンネルから製品証明書を取得するまで、デフォルトの製品証明書が使用されます。

2.8. クラスターリングと高可用性

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされたクラスターリングおよび高可用性の変更の概要について説明しています。

2.8.1. Luci 置換制限 (pcs)

Red Hat Enterprise Linux 6 では、**luci** が Red Hat Enterprise Linux 5 と Red Hat Enterprise Linux 6 の両方の高可用性クラスターを制御していました。

Red Hat Enterprise Linux 7 では、**pcs** に置き換えられ、**luci** は削除されました。**pcs** で制御できるのは、Red Hat Enterprise Linux 7 の Pacemaker ベースのクラスターのみです。Red Hat Enterprise Linux 6 rgmanager ベースの高可用性クラスターは制御できません。

2.8.2. Piranha に代わる Keepalived

Red Hat Enterprise Linux 7 の Load Balancer Add-On には、**keepalived** サービスが追加されました。このアドオンにより、**piranha** で使用可能な機能および追加機能の両方が提供されます。したがって、**piranha** は、Red Hat Enterprise Linux 7 の **keepalived** サービスに置き換えられました。

併せて、設定ファイルとその形式が変更になりました。**keepalived** は、デフォルトで `/etc/keepalived/keepalived.conf` ファイルで設定されます。このファイルで使用する設定フォーマットおよび構文の詳細は、**keepalived.conf** の man ページに記載してあります。

```
$ man keepalived.conf
```

2.8.3. オンライン移行の制限

Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 へのオンライン移行では、クラスターがサポート対象外となっています。

また、Red Hat Enterprise Linux 6 の高可用性スタックは、Red Hat Enterprise Linux 7 の高可用性スタックと互換性がありません。したがって、Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 への高可用性クラスターのオンライン移行はサポートされていません。

2.8.4. 新リソースマネージャー (Pacemaker)

Red Hat Enterprise Linux 7 では、**rgmanager** と **cman** が、**pacemaker** および **corosync** に置き換えられています。

Pacemaker は、多くの便利な機能を持つ高可用性リソースマネージャーです。

- マシンおよびアプリケーションレベルの障害を検出して回復。
- 多くの冗長設定をサポート。
- 定足数およびリソース駆動型クラスターをサポート。
- クォーラムが失われる場合の処理に対する設定可能な戦略 (複数マシンの失敗時)。
- アプリケーションがどのマシンにあるかに関わらず、アプリケーションのスタートアップおよびシャットダウンの順位付けの指定をサポート。
- アプリケーションが同一マシンで稼働する必要があるかないかについての指定をサポート。

- アプリケーションを複数マシン上でアクティブにすることの指定をサポート。
- マスターやスレーブのようなアプリケーションの複数モードをサポート。
- どの失敗やクラスター状態にも立証可能な正しい反応。
- 状況が発生する前に、その状況に対する反応がオフラインでテスト可能。

Pacemaker の詳細については、[高可用性アドオンのドキュメント](#)を参照してください。

2.8.5. 新機能: リソースエージェント

Red Hat Enterprise Linux 7 では、Pacemaker リソースマネージャーと連携するリソースエージェントが導入されています。リソースエージェントは、クラスターリソースを概念化し、クラスター環境でリソースを管理する標準インターフェイスを提供します。Red Hat Enterprise Linux 7 で利用可能なリソースエージェントの詳細については、[高可用性アドオンのドキュメント](#)を参照してください。

DB2 を高可用性環境のクラスターリソースとして使用して管理する IBM DB2 リソースエージェントのサポートが Red Hat Enterprise Linux 7.2 に追加されています。

2.8.6. quorum 実装の変更

Red Hat Enterprise Linux 6 に同梱されていた `qdiskd` は Red Hat Enterprise Linux 7 では削除されました。新たな quorum 実装は、`corosync` パッケージに含まれる `votequorum` で提供されており、ほとんどのユースケースで `qdiskd` に代わるように拡張されています。拡張子 (`wait_for_all`、`auto_tie_breaker`、`last_man_standing`) は `votequorum.5` man ページに詳細に記載されています。

```
$ man 5 votequorum
```

2.9. デスクトップ

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされたサポート対象のデスクトップユーザー環境の変更概要について説明しています。

このセクションでは、Red Hat Enterprise Linux 7 での新デスクトップ環境でユーザーが予期する主な変更点のみ説明されています。詳細については、[デスクトップ移行および管理ガイド](#)を参照してください。

2.9.1. 新たなデフォルトのデスクトップ環境 (GNOME クラシック)

GNOME クラシックは、Red Hat Enterprise Linux 7 における GNOME 3 デスクトップ環境のデフォルトセッションです。この環境は、GNOME 3 デスクトップ環境への拡張セットとして提供されており、慣れ親しんだ GNOME 2 のルックアンドフィールを維持している一方で、強力な新機能が含まれています。

GNOME クラシックのユーザーインターフェイスは、以下の 2 つの主要コンポーネントで設定されています。

トップバー

画面最上部にあるこのバーは、アプリケーションと場所メニューを表示します。

アプリケーションメニューからは、システム上のアプリケーションにアクセスできます。アプリケーションは、いくつかのカテゴリーに分けられています。このメニューからは新しい **アクティビティー画面** にもアクセスでき、ここでは開いているウィンドウ、ワークスペース、メッセージ、シ

システム通知を簡単に見ることができます。

場所メニューは、トップバーのアプリケーションメニューの横に表示されます。ユーザーは、ここから **ダウンロード**、**写真** などの重要なディレクトリーに簡単にアクセスできます。

タスクバー

タスクバーは画面の一番下に表示され、ウィンドウ一覧や通知アイコン、現在のワークスペースのショート ID、利用可能なワークスペースの合計数を表示します。

GNOME クラシックとその機能の完全なガイド、および Red Hat Enterprise Linux 7 で利用可能なその他のデスクトップ環境については、[デスクトップ移行および管理ガイド](#)を参照してください。

2.9.2. 新デスクトップ環境 (GNOME 3)

Red Hat Enterprise Linux 7 では、GNOME 3 デスクトップ環境の GNOME 3 セッションもサポートしています。この環境は、使いやすく、ユーザーの生産性を高める設計になっています。オンラインドキュメントストレージサービス、カレンダー、連絡先とのすぐれた統合機能を提供しているので、いつも最新情報にアクセスできます。

GNOME 3 のユーザーインターフェイスは 3 つの主要コンポーネントで設定されています。

トップバー

画面最上部にあるこの水平のバーからは、**アクティビティー画面**、時計、カレンダー、システムステータスアイコン、システムメニューなど、GNOME Shell の基本的な機能にアクセスできます。

アクティビティー画面

アクティビティー画面 では、開いているウィンドウやワークスペース、メッセージ、システム通知を簡単に見ることができます。検索バーは、ファイルをみつけたり、アプリケーションを起動したり、設定ツールを開いたりするのに最も簡単な方法です。左側のダッシュにはお好みのアプリケーションが表示されるので、よく使うツールにすばやくアクセスすることができます。

メッセージトレイ

メッセージトレイは、画面の一番下の水平バーに表示されます。保留中の通知を表示するので、常にシステムに何が起きているのかが分かります。

GNOME 3 とその機能の完全なガイド、および Red Hat Enterprise Linux 7 で利用可能なその他のデスクトップ環境については、[デスクトップ移行および管理ガイド](#)を参照してください。

2.9.3. KDE Plasma ワークスペース (KDE)

Red Hat Enterprise Linux 7 は、KDE Plasma ワークスペース (KDE) バージョン 4.10 を提供しています。これは、以前は K デスクトップ環境と呼ばれていました。KDE のこの更新バージョンには、以下のものを含む多くの機能強化が含まれています。

- 洗練された一貫性のあるルックアンドフィールのデフォルトの Oxygen スタイル。
- パネルに進行状況が視覚的に表示される更新版通知システム (スピードグラフを伴う移動かつ閉鎖可能な通知)。
- ワークスペース設定が **システム設定** で可能になりました。
- **Activity Manager** がアクティビティーを追加、削除、保存、復元でき、アクティビティー間でスイッチする機能を提供します。

- コアおよびユーザーインターフェイス要素が最適化され、パフォーマンスが向上します。
- シンプルなユーザーインターフェイスと容易なプロファイルスイッチングのある適応型電源管理。
- プリンター設定を簡素化し、正確な印刷ステータスをすばやく提供する新しい **Print Manager**。
- ナビゲーションボタンやタブブラウズが可能で、メタデータ処理が改善された更新版 **Dolphin File Manager**。
- タブ、ウィンドウコントロール、および相互運用性が改善された更新版端末エミュレーター (**Konsole**)。
- 新たなディスプレイマネージャーの **KScreen**。これは、解像度や相対的位置を含むディスプレイ設定を自動的に記憶し、復元できるものです。
- 新アプレットの **Plasma Network Manager** は、ネットワーク制御とネットワーク接続設定を容易にします。

ただし、**Kmail** は Red Hat Enterprise Linux 7 に含まれていないことに注意してください。

2.10. 開発者ツール

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされた開発者ツール変更の概要について説明しています。

2.10.1. Red Hat Developer Toolset

Red Hat Developer Toolset は、個別の高速ライフサイクルで、最新バージョンの安定したオープンソース開発者ツールを提供します。Red Hat では、アクティブな Red Hat Developer サブスクリプションをお持ちの場合にご利用いただけます。

Red Hat Developer Toolset 2 は現在、Red Hat Enterprise Linux 7 上でのアプリケーション開発をサポートしていません。ただし、Red Hat Developer Toolset は Red Hat Enterprise Linux 6 上でのアプリケーション開発をサポートしており、これらのアプリケーションは、サポート対象の Red Hat Enterprise Linux 6 のマイナーリリースもしくは Red Hat Enterprise Linux 7 で導入できます。

2.10.2. 互換性ライブラリー

Red Hat Enterprise Linux 7 には、以前のリリースの Red Hat Enterprise Linux からのインターフェイスをサポートする互換性ライブラリーが同梱されます。このライブラリーは、Red Hat の互換性ポリシーおよび Red Hat の判断に従って同梱されます。詳細については、[アプリケーション互換性ガイド](#)を参照してください。

以下の互換性ライブラリーが Red Hat Enterprise Linux 7 に含まれています。

表2.5 互換性ライブラリー

ライブラリー	対象インターフェイスがデフォルトとなっている最新リリース
compat-db47	Red Hat Enterprise Linux 6

ライブラリー	対象インターフェイスがデフォルトとなっている最新リリース
compat-libcap1	Red Hat Enterprise Linux 5
compat-libf2c-34	Red Hat Enterprise Linux 4
compat-libgfortran-41	Red Hat Enterprise Linux 5
compat-openldap	Red Hat Enterprise Linux 5
libpng12	Red Hat Enterprise Linux 5
openssl098e	Red Hat Enterprise Linux 5
compat-dapl	Red Hat Enterprise Linux 5
compat-libtiff3	Red Hat Enterprise Linux 6
compat-libstdc++-33	Red Hat Enterprise Linux 3 (オプションのリポジトリのみ)

Red Hat Enterprise Linux 7 には、**compat-gcc-44** および **compat-gcc-44-c++** パッケージも含まれています。これらは Red Hat Enterprise Linux 6 に同梱されたシステムコンパイラーを表すもので、**compat-glibc** パッケージとともに、レガシーソフトウェアの構築およびリンクに使用することができます。

2.11. セキュリティーおよびアクセス制御

このセクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 との間でなされたセキュリティ、アクセス制御、および関連設定ツールの変更の概要について説明しています。

2.11.1. 新ファイアウォール (firewalld)

Red Hat Enterprise Linux 6 では、**iptables** ユーティリティーがファイアウォール機能を提供し、コマンドラインもしくはグラフィカル設定ツールの **system-config-firewall** で設定されていました。Red Hat Enterprise Linux 7 では、**iptables** がファイアウォール機能を提供しています。ただし、管理者は動的ファイアウォールデーモンである **firewalld** と、その設定ツールである **firewall-config**、**firewall-cmd**、**firewall-applet** で **iptables** と通信します。これは、Red Hat Enterprise Linux 7 のデフォルトインストールには含まれていません。

firewalld は動的であることから、その設定はいつでも変更可能で、即座に実行されます。ファイアウォールはリロードする必要がないことから、既存のネットワーク接続で意図しない中断が発生することはありません。

Red Hat Enterprise Linux 6 と 7 間でのファイアウォールの主な相違点は以下のとおりです。

- **Firewalld** 設定の詳細は **/etc/sysconfig/iptables** に保存されていません。設定詳細は **/usr/lib/firewalld** および **/etc/firewalld** ディレクトリーの様々なファイルに保存されます。

- Red Hat Enterprise Linux 6 では、設定が変更される度にすべてのルールが削除され、再適用されていましたが、**firewalld** は設定の差異のみを適用します。その結果、**firewalld** は既存の接続を中断することなく、ランタイム中に設定を変更することができます。

Red Hat Enterprise Linux 7 でファイアウォールを設定するための追加情報と支援については、[セキュリティガイド](#)を参照してください。

2.11.1.1. firewalld への移行ルール



重要

Red Hat Enterprise Linux 7 を、別の Red Hat 製品 (Red Hat Enterprise Linux OpenStack Platform など) と使用している場合は、**firewalld** に移行する代わりに **iptables** または **ip6tables** を引き続き使用することが適切なことがあります。

どのファイアウォールユーティリティーを使用すればいいかわからない場合は、製品ドキュメントを参照するか、Red Hat サポートにお問い合わせください。

firewalld を無効にし、**iptables** または **ip6tables** を引き続き使用する方法については、<https://access.redhat.com/articles/1229233> を参照してください。

Red Hat Enterprise Linux 6 では、以下の 2 つの方法でファイアウォールを設定していました。

- グラフィカルな **system-config-firewall** ツールを使ってルールを設定。このツールは、設定詳細を `/etc/sysconfig/system-config-firewall` ファイルに保存し、`/etc/sysconfig/iptables` ファイルに **iptables** サービスを、および `/etc/sysconfig/ip6tables` ファイルに **ip6tables** サービスを設定していました。
- 手動で `/etc/sysconfig/iptables` ファイルおよび `/etc/sysconfig/ip6tables` ファイルを編集 (まったくのゼロから、もしくは **system-config-firewall** が作成した初期設定を編集)。

Red Hat Enterprise Linux 6 のファイアウォールを **system-config-firewall** で設定している場合、システムをアップグレードして **firewalld** をインストールした後に、**firewall-offline-cmd** ツールを使って、`/etc/sysconfig/system-config-firewall` の設定を **firewalld** のデフォルトゾーンに移行することができます。

```
$ firewall-offline-cmd
```

ただし、`/etc/sysconfig/iptables` もしくは `/etc/sysconfig/ip6tables` を手動で作成または編集している場合は、**firewalld** のインストール後に **firewall-cmd** または **firewall-config** で新しい設定を作成するか、**firewalld** を無効にして旧型の **iptables** および **ip6tables** サービスの使用を継続する必要があります。新しい設定の作成または **firewalld** の無効化の詳細については、[セキュリティガイド](#)を参照してください。

2.11.2. PolicyKit の変更

これまで、PolicyKit は、**.pkla** ファイル内のキーの値のペアを使って追加のローカル権限を定義してきました。Red Hat Enterprise Linux 7 では、JavaScript を使ってローカル権限を定義する機能が提供され、必要に応じて権限を書くことが可能になっています。

polkitd は、**.rules** ファイルを辞書式順序で、`/etc/polkit-1/rules.d` および `/usr/share/polkit-1/rules.d` ディレクトリーから読み込みます。2 つのファイルが同じ名前を共有している場合は、`/etc` にあるファイルが `/usr` にあるファイルよりも先に処理されます。以前の **.pkla** ファイルでは、最後に処理されたルールが優先されていました。新たな **.rules** ファイルでは、最初に合致するルールが優先されます。

移行後、既存ルールは、`/etc/polkit-1/rules.d/49-polkit-pkla-compat.rules` ファイルによって適用されます。このため、`.rules` ファイルが `/usr` または `/etc` にあり、かつそのファイル名が辞書式順序で **49-polkit-pkla-compat** の先にくる場合は、既存ルールよりも優先されます。古いルールが無効にならないようにする一番簡単な方法は、他の全 `.rules` ファイルの名前を、49 よりも大きい番号で始めることです。

詳細については、[デスクトップ移行および管理ガイド](#)を参照してください。

2.11.3. ユーザー ID の変更

Red Hat Enterprise Linux 6 でのベースユーザー ID は **500** でした。Red Hat Enterprise Linux 7 でのベースユーザー ID は **1000** となっています。この変更にしたがい、アップグレードプロセス中に `/etc/login.defs` ファイルが置き換えられます。

デフォルトの `/etc/login.defs` ファイルを修正していない場合、このファイルはアップグレード中に置き換えられます。ベースユーザー ID の番号が **1000** に変更になり、新規ユーザーに割り当てられるユーザー ID は、1000 またはそれ以上になります。この変更前に作成されたユーザーアカウントは、現行のユーザー ID を維持し、期待通りに機能し続けます。

デフォルトの `/etc/login.defs` ファイルを修正している場合は、このファイルはアップグレード中に置き換えられず、ベースユーザー ID 番号は 500 のままになります。

2.11.4. libuser の変更

Red Hat Enterprise Linux 7 では、**libuser** ライブラリーは **ldap** および **files** モジュールの両方を含む設定、もしくは **ldap** および **shadow** モジュールの両方を含む設定をサポートしません。これらのモジュールを組み合わせるとパスワード処理に曖昧さが発生するので、そのような設定は初期化プロセス中に拒否されるようになっています。

LDAP のユーザーもしくはグループの管理に **libuser** を使用する場合は、**files** モジュールおよび **shadow** モジュールを、設定ファイル (デフォルトでは `/etc/libuser.conf`) の **modules** ディレクティブおよび **create_modules** ディレクティブから削除する必要があります。

2.11.5. opencryptoki キーストアの変更

以前のバージョンの Red Hat Enterprise Linux は、**opencryptoki** キーストアのバージョン 2 を使用していました。このバージョンでは、ハードウェアでセキュアキーを使用して、プライベートトークンを暗号化していました。Red Hat Enterprise Linux 7 では、ソフトウェアのクリアキーでプライベートトークンオブジェクトを暗号化するバージョン 3 が使用されます。したがって、バージョン 2 で作成したプライベートのトークンオブジェクトをバージョン 3 で使用するには、最初にバージョン 2 で作成したトークンオブジェクトを移行する必要があります。

プライベートトークンオブジェクトを移行するには、以下の手順を実行します。

1. **opencryptoki** のバージョンが最新であることを確認します。

```
# yum update -y opencryptoki
```

2. トークンのスロット数を確認するには、**pkcsconf** でトークンのスロット数を調べます。以下のコマンドを root 権限で実行します。

```
# pkcsconf -s
# pkcsconf -t
```

トークンのスロット番号に注意してください。スロットの説明は **(CCA)** で終わります。情報フィールドは、このトークンを **IBM CCA トークン** として識別します。

3. インターフェイスへのアクセスを停止するには、**pkcsslotd** サービスおよびすべての **opencryptoki** プロセスを停止します。

```
# systemctl stop pkcsslotd.service
```

以下のコマンドを実行して **kill** ユーティリティーを停止するプロセスを特定し、適切なプロセスを終了します。

```
# ps ax | grep pkcsslotd
```

4. 移行前に、CCA データストア (トークンが保存されているディレクトリー、通常は **/var/lib/opencryptoki/ccatok**) のバックアップを取得します。たとえば、このファイルのコピーを作成します。

```
# cp -r /var/lib/opencryptoki/ccatok /var/lib/opencryptoki/ccatok.backup
```

5. 移行ユーティリティーを実行し、**/var/lib/opencryptoki/ccatok** ディレクトリーに変更し、移行ユーティリティーを実行します。

```
# cd /var/lib/opencryptoki/ccatok
# pkcscca -m v2objectsv3 -v
```

要求されたら、セキュリティーオフィサー (SO) PIN とユーザー PIN を入力します。

6. 古い共有メモリーファイルを削除すると、手動で **/dev/shm/var.lib.opencryptoki.ccatok** ファイルを削除するか、システムを再起動します。

```
# rm /dev/shm/var.lib.opencryptoki.ccatok
```

7. 操作インターフェイスアクセスに戻ります。再度、**pkcsslotd** サービスを起動します。

```
# systemctl start pkcsslotd.service
```

移行の際に問題が発生した場合は、以下を確認してください。

- コマンドを **root** で実行しており、**root** が **pkcs11** グループのメンバーになっていること。
- **pkcsconf** ユーティリティーが **/usr/lib/pkcs11/methods/** ディレクトリーまたは **/usr/sbin/** ディレクトリーのいずれかにあること。
- トークンのデータストアが **/var/lib/opencryptoki/ccatok/** ディレクトリーにあること。
- スロット番号が指定されており、その番号が正しいこと。
- セキュリティーオフィス (SO) PIN およびユーザー PIN が正しいことを確認します。
- カレントディレクトリーに書き込み権限があること。

第3章 パッケージ、機能、およびサポートの変更

このセクションでは、Red Hat Enterprise Linux 7 で提供される機能またはパッケージの変更、およびそのパッケージのサポートの変更について説明しています。

3.1. 新しいパッケージ

このセクションでは、Red Hat Enterprise Linux 7 で利用可能な、注目すべきパッケージについて説明しています。

3.1.1. Preupgrade Assistant

Preupgrade Assistant (preupg) は、使用中のシステムに変更を加える前に、Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 へのアップグレード時に直面する可能性のある問題をチェックします。これにより、実際にアップグレードプロセスを開始する前に、Red Hat Enterprise Linux 7 へのアップグレードが成功する可能性を評価することができます。

Preupgrade Assistant は、パッケージの削除、互換性のない古い機能、名前の変更、設定ファイルの互換性欠如など、システムにインプレースアップグレードを行った場合に考えられる限界を評価します。次に、以下を提供します。

- 検出された移行問題に対する解決案を含むシステム分析レポート
- システムのクローンを作成する際に利用できるデータ (インプレースアップグレードに適していない場合)
- インプレースアップグレードの終了後、より複雑な問題を解決するためのポストアップグレードスクリプト

Preupgrade Assistant によって保存される情報およびログ記録以外、システムに一切の変更は加えられません。

Preupgrade Assistant の取得および使用に関する詳細な指示については、[アップグレードの適合性の評価](#) を参照してください。

3.1.2. Red Hat Upgrade Tool

Preupgrade Assistant の後に新しい **Red Hat Upgrade Tool** が使用され、以下の3つのアップグレードプロセスフェーズを処理します。

- **Red Hat Upgrade Tool** が、ディスクまたはサーバーからパッケージとアップグレードイメージを取り込み、システムをアップグレード用に準備して再起動します。
- システムが再起動後にアップグレードパッケージが利用可能であることを検出し、**systemd** および **yum** を使用してシステム上のパッケージをアップグレードします。
- アップグレード後に、**Red Hat Upgrade Tool** がクリーンアップを実行し、アップグレードしたオペレーティングシステムに再起動します。

ネットワークベースおよびディスクベースのアップグレードがサポートされています。システムアップグレードの詳細な説明については、[1章 アップグレード方法](#) を参照してください。

3.1.3. Chrony

Chrony は、**chrony** パッケージで提供される新 NTP クライアントです。Red Hat Enterprise Linux 7 の

デフォルト NTP として、参照実装 (`ntp`) に代わるものです。ただし、`ntp` で利用可能な機能をすべてサポートするわけではないため、互換性の理由で `ntp` が提供されています。`ntp` が必要な場合は、`chrony` を明示的に削除して、`ntp` をインストールする必要があります。

Chrony の時間管理アルゴリズムは、`ntp` 実装に比べていくつかの利点があります。

- 同期がより速く正確。
- 周波数訂正の幅がより広い。
- 時計の周波数の急激な変更によく対応。
- 初期同期の後のクロックステップがない。
- 断続的なネットワーク接続でうまく機能。

`chrony` の詳細については、[システム管理者ガイド](#)を参照してください。

3.1.4. HAProxy

HAProxy は、高可用性環境に適した TCP/HTTP リバースプロキシです。リソースをほとんど必要とせず、イベント駆動型のアーキテクチャーにより、システムの安定性をリスクにさらすことなく、数百ものインスタンス上で、同時に数千もの接続を容易に処理できます。

HAProxy に関する詳細は、`man` ページを参照するか、`/usr/share/doc/haproxy` ディレクトリーにある、`haproxy` パッケージからインストールされたドキュメントを参照してください。

3.1.5. Kernel-tools

`kernel-tools` パッケージには、Linux カーネル用のツールが多数含まれています。このパッケージ内のツールのいくつかは、他のパッケージでこれまで利用可能だったツールに代わるものです。詳細は、「[非推奨パッケージ](#)」および「[パッケージの置換](#)」を参照してください。

3.1.6. NFQUEUE (libnetfilter_queue)

Red Hat Enterprise Linux 7.1 では、`libnetfilter_queue` パッケージが提供されます。このライブラリーを使用すると、**NFQUEUE** iptables ターゲットが有効になります。この場合、リッスン中のユーザー領域アプリケーションが、指定されたキューからパケットを取得し、そのパケットの処理方法を決定するよう指定します。

3.1.7. SCAP セキュリティーガイド

`scap-security-guide` パッケージは、セキュリティガイドンス、ベースライン、および Security Content Automation Protocol (SCAP) に関連する検証メカニズムを提供します。以前は、このパッケージは EPEL リポジトリー (Enterprise Linux 用追加パッケージ) でのみ利用できていました。Red Hat Enterprise Linux 7.1 より、`scap-security-guide` は Red Hat Enterprise Linux 7 Server (RPMS) リポジトリーで利用できます。

3.1.8. Red Hat Access GUI

Red Hat Access GUI は、Red Hat ナレッジベース、リソース、および機能を使用してヒント、回答、診断サービスを利用できるデスクトップアプリケーションです。[Red Hat カスタマーポータル](#) でアクティブなアカウントをお持ちの場合は、ナレッジベースをキーワードで簡単に検索して追加情報およびヒントにアクセスできます。GNOME Desktop のインストールを選択している場合は、**Red Hat Access GUI** はすでにインストールされています。

このツールの利点、インストールおよび使用方法については、[Red Hat Access GUI](#) を参照してください。

3.2. パッケージの置換

このセクションでは、バージョン 6 から 7 の間に Red Hat Enterprise Linux から削除されたパッケージの一覧を紹介します。また、Red Hat Enterprise Linux 7 で同等の機能を持つ置換パッケージまたは代替パッケージが利用可能な場合は併記します。

表3.1 置換パッケージ

削除されたパッケージ	置換/代替	備考
vconfig	iproute (ip ツール)	完全に互換性があるわけではありません。
module-init-tools	kmod	
openoffice.org	libreoffice	
man	man-db	
ext2 および ext3 filesystem driver	ext4 filesystem driver	
openais	corosync	機能は、Red Hat Enterprise Linux HA スタックに含まれています。
jwhois	whois	出力フォーマットが異なります。
libjpeg	libjpeg-turbo	
gpxe	ipxe	gpxe の分岐。
cpuspeed	kernel、kernel-tools (cpupower、cpupower.service)	/etc/sysconfig/cpupower で設定されるようになりました。ユーザースペースのサイズ変更デーモンは含まれなくなりました。必要な場合は、カーネルガバナーを使用してください。
nc	nmap-ncat	
procps	procps-ng	
openswan	libreswan	
arptables_jf	arptables	

削除されたパッケージ	置換/代替	備考
gcj	OpenJDK	Java アプリケーションを、gcj を使ってネイティブコードにコンパイルしないでください。
インストールアーキテクチャーとしての 32-bit x86	AMD64 および Intel 64	アプリケーションは、互換性ライブラリーを使って実行されます。64 ビットの Red Hat Enterprise Linux 6 上でアプリケーションをテストしてください。32 ビットの x86 起動サポートが必要な場合は、引き続き Red Hat Enterprise Linux 6 を使用してください。
Power 6 PPC support		引き続き Red Hat Enterprise Linux 5 または Red Hat Enterprise Linux 6 を使用してください。
Matahari	CIM ベースの管理	
ecryptfs	既存の LUKS または dm-crypt ブロックベースの暗号化を使用してください。	暗号化ファイルシステムは移行できません。暗号化されたデータは再作成する必要があります。
evolution-exchange	evolution-mapi/evolution-ews	
TurboGears2 web application stack		
openmotif22	motif	現行の Motif バージョンに対してアプリケーションを再構築してください。
webalizer web analytics tool		他の web アナリティクスツールの方が優れています。
compiz window manager	gnome-shell	
Eclipse developer toolset		Eclipse は、Developer Toolset オファリングで提供されるようになりました。
Qpid および QMF		Qpid および QMF は、MRG オファリングで利用可能です。
amtu		コモンクライテリア認定では、このツールは不要になりました。

削除されたパッケージ	置換/代替	備考
pidgin frontends	empathy	
perl-suidperl	perl	この機能は、アップストリームの perl では削除されました。
pam_passwdqc、pam_cracklib	libpwquality、pam_pwquality	完全に互換性があるわけではありません。
HAL ライブラリーおよびデーモン	udev	
ConsoleKit ライブラリーおよびデーモン	systemd	完全に互換性があるわけではありません。
system-config-network	nm-connection-editor、nmcli	
thunderbird	evolution	
system-config-firewall	firewalld	
busybox	通常のユーティリティー	
KVM および virt パッケージ (ComputeNode 内)	サーバーバリエーションなど、KVM および virt が備わったバリエーション	
abyssinica-fonts	sil-abyssinica-fonts	
axis	java-1.7.0-openjdk	
ccs	pcs	完全に互換性があるわけではありません。
ckuni-fonts-common	ckuni-uming-fonts	
classpath-jaf	java-1.7.0-openjdk	
classpath-mail	javamail	完全に互換性があるわけではありません。
cman	corosync	
control-center-extra	control-center	
db4-cxx	libdb4-cxx	
db4-devel	libdb4-devel	

削除されたパッケージ	置換/代替	備考
db4-utils	libdb4-utils	
desktop-effects	control-center	
DeviceKit-power	upower	完全に互換性があるわけではありません。
dracut-kernel	dracut	
eggdbus	glib2	完全に互換性があるわけではありません。
fcoe-target-utils	targetcli	詳細は、「 targetcli によるターゲット管理 」を参照してください。
febootstrap	supermin	
gcc-java	java-1.7.0-openjdk-devel	
GConf2-gtk	GConf2	
gdm-plugin-fingerprint	gdm	
gdm-plugin-smartcard	gdm	
gdm-user-switch-applet	gnome-shell	完全に互換性があるわけではありません。
geronimo-specs	geronimo-parent-poms	
geronimo-specs-compatible	geronimo-jms、geronimo-jta	完全に互換性があるわけではありません。
gimp-help-browser	gimp	完全に互換性があるわけではありません。
gnome-applets	gnome-classic-session	完全に互換性があるわけではありません。
gnome-keyring-devel	gnome-keyring	
gnome-mag	gnome-shell	完全に互換性があるわけではありません。

削除されたパッケージ	置換/代替	備考
gnome-python2-applet	pygtk2	完全に互換性があるわけではありません。
gnome-speech	speech-dispatcher	完全に互換性があるわけではありません。
gpxe-roms-qemu	ipxe-roms-qemu	
hal	systemd	完全に互換性があるわけではありません。
hal-devel	systemd-devel	完全に互換性があるわけではありません。
ibus-gtk	ibus-gtk2	
ibus-table-cangjie	ibus-table-chinese-cangjie	
ibus-table-erbi	ibus-table-chinese-erbi	
ibus-table-wubi	ibus-table-chinese-wubi-haifeng	
jakarta-commons-net	apache-commons-net	
java-1.5.0-gcj	java-1.7.0-openjdk、java-1.7.0-openjdk-headless	完全に互換性があるわけではありません。
java-1.5.0-gcj-devel	java-1.7.0-openjdk-devel	完全に互換性があるわけではありません。
java-1.5.0-gcj-javadoc	java-1.7.0-openjdk-javadoc	完全に互換性があるわけではありません。
junit4	junit	
jwhois	whois	
kabi-whitelists	kernel-abi-whitelists	
kdeaccessibility-libs	kdeaccessibility	
kdebase-devel	kde-baseapps-devel	
kdebase-workspace-wallpapers	kde-wallpapers	
kdelibs-experimental	kdelibs	

削除されたパッケージ	置換/代替	備考
kdesdk-libs	kate-libs、kdesdk-kmtrace-libs、kdesdk-kompare	完全に互換性があるわけではありません。
kdesdk-utils	kdesdk-poxml	
krb5-auth-dialog	gnome-online-accounts	完全に互換性があるわけではありません。
lldpad-libs	lldpad	
lsik	util-linux	完全に互換性があるわけではありません。
luci	pcs	詳細は、「 クラスタリングと高可用性 」を参照してください。
man-pages-uk	man-pages	
mingetty	util-linux	完全に互換性があるわけではありません。
modcluster	pcs	完全に互換性があるわけではありません。
mod_perl	mod_fcgid	httpd 2.4 と互換性なし
m17n-contrib-*	m17n-contrib	
m17n-db-*	m17n-db、m17n-db-extras	
NetworkManager-gnome	nm-connection-editor、network-manager、applet	
nss_db	glibc	完全に互換性があるわけではありません。
openais	corosync	
openaislib	corosynclib	
openaislib-devel	corosynclib-devel	
PackageKit-gtk-module	PackageKit-gtk3-module	完全に互換性があるわけではありません。
polkit-desktop-policy	polkit	

削除されたパッケージ	置換/代替	備考
pulseaudio-libs-zeroconf	pulseaudio-libs	完全に互換性があるわけではありません。
qt-sqlite	qt	
rdesktop	xfreerdp	
Red_Hat_Enterprise_Linux-Release_Notes-6-*	Red_Hat_Enterprise_Linux-Release_Notes-7-*	
redhat-lsb-compat	redhat-lsb-core	
rgmanager	pacemaker	詳細は、「 クラスタリングと高可用性 」を参照してください。
rhythmbox-upnp	rhythmbox	
ricci	pcs	詳細は、「 クラスタリングと高可用性 」を参照してください。
samba4*	samba*	詳細は、「 Samba 」を参照してください。
sbm-cim-client	sbm-cim-client2	完全に互換性があるわけではありません。
scsi-target-utils	targetcli	詳細は、「 targetcliによるターゲット管理 」を参照してください。
seekwatcher	iowatcher	
spice-client	virt-viewer	完全に互換性があるわけではありません。
system-config-lvm	gnome-disk-utility	完全に互換性があるわけではありません。
texlive-*	texlive	
tex-cm-lgc	texlive-cm-lgc	
tex-kerkis	texlive-kerkis	
texlive-texmf-dvips	texlive-dvips	

削除されたパッケージ	置換/代替	備考
texlive-texmf-latex	texlive-latex	
tomcat6	tomcat	
tomcat6-el-2.1-api	tomcat-el-2.2-api	
tomcat6-jsp-2.1-api	tomcat-jsp-2.2-api	
tomcat6-lib	tomcat-lib	
totem-upnp	totem	
udisks	udisks2	完全に互換性があるわけではありません。
un-core-batang-fonts	nhn-nanum-myeongjo-fonts	
un-core-dinaru-fonts、un-core-graphic-fonts	nhn-nanum-gothic-fonts	完全に互換性があるわけではありません。
un-core-dotum-fonts	nhn-nanum-gothic-fonts	
un-core-fonts-common	nhn-nanum-fonts-common	完全に互換性があるわけではありません。
un-core-gungseo-fonts	nhn-nanum-brush-fonts	完全に互換性があるわけではありません。
un-core-pilgi-fonts	nhn-nanum-pen-fonts	完全に互換性があるわけではありません。
unique	unique3、glib2	完全に互換性があるわけではありません。
unique-devel	unique3-devel	完全に互換性があるわけではありません。
unix2dos	dos2unix	
vgabios	seavgabios-bin	
w3m	text-www-browser	完全に互換性があるわけではありません。
xmlrpc3-*	xmlrpc-*	

削除されたパッケージ	置換/代替	備考
xorg-x11-drv-apm	xorg-x11-drv-fbdev、xorg-x11-drv-vesa	
xorg-x11-drv-ast、xorg-x11-drv-cirrus、xorg-x11-drv-mga	xorg-x11-drv-modesetting	
xorg-x11-drv-ati-firmware	linux-firmware	
xorg-x11-drv-elographics、xorg-x11-drv-glint、xorg-x11-drv-i128、xorg-x11-drv-i740、xorg-x11-drv-mach64、xorg-x11-drv-rendition、xorg-x11-drv-r128、xorg-x11-drv-savage、xorg-x11-drv-siliconmotion、xorg-x11-drv-sis、xorg-x11-drv-sisusb、xorg-x11-drv-s3virge、xorg-x11-drv-tdfx、xorg-x11-drv-trident、xorg-x11-drv-vooodoo、xorg-x11-drv-xgi	xorg-x11-drv-fbdev、xorg-x11-drv-vesa	
xorg-x11-drv-nv	xorg-x11-drv-nouveau	
xorg-x11-twm	metacity	完全に互換性があるわけではありません。
xorg-x11-xdm	gdm	完全に互換性があるわけではありません。
yum-plugin-downloadonly	yum	

3.3. 非推奨パッケージ

このセクションで挙げられているパッケージは、Red Hat Enterprise Linux 7 では非推奨とみなされています。これらのパッケージは依然として機能し、サポートされますが、Red Hat ではその使用を推奨していません。

表3.2 非推奨パッケージ

機能/パッケージ	代替	移行における注意
ext2 ファイルシステムのサポート	ext3、ext4	ext4 を ext2 および ext3 ファイルシステムに使用できます。
sblim-sfcb	tog-pegasus	

機能/パッケージ	代替	移行における注意
レガシーの RHN がホストする登録	subscription-manager および Subscription Asset Manager	
acpid	systemd	
evolution-mapi	evolution-ews	Microsoft Exchange Server 2003 マシンからの移行を行なってください。
gtkhtml3	webkitgtk3	
sendmail	postfix	
edac-utils および mcelog	rasdaemon	
libcgroup	systemd	cgutils は引き続き Red Hat Enterprise Linux 7.0 に残りますが、今後のリリースで systemd に移行できるように機能は進化しています。
lvm1	lvm2	
lvm2mirror および cmirror	lvm2 raid1	

3.4. 削除済みパッケージ

以下のパッケージは、Red Hat Enterprise Linux バージョン 6 とバージョン 7 との間で削除され、サポート対象外となっています。これらのパッケージのなかには、同等の機能を持つ代替パッケージがあるものもあります。詳細は、「[パッケージの置換](#)」を参照してください。

- amtu
- ant-antlr
- ant-apache-bcel
- ant-apache-bsf
- ant-apache-log4j
- ant-apache-oro
- ant-apache-regexp
- ant-apache-resolver
- ant-commons-logging
- ant-commons-net

- ant-javamail
- ant-jdepend
- ant-jsch
- ant-junit
- ant-nodeps
- ant-swing
- ant-trax
- apache-jasper
- apache-tomcat-apis
- apr-util-ldap
- arts
- arts-devel
- aspell
- atmel-firmware
- at-spi
- at-spi-python
- audiofile
- audit-viewer
- avahi-tools
- avahi-ui
- avalon-framework
- avalon-logkit
- batik
- brasero
- brasero-libs
- brasero-nautilus
- bsf
- busybox
- b43-fwcutter

- b43-openfwfwf
- cas
- cdparanoia
- cdrdao
- cjet
- cloog-ppl
- cluster-cim
- cluster-glue
- cluster-glue-libs
- cluster-glue-libs-devel
- clusterlib
- clusterlib-devel
- cluster-snmp
- cman
- compat-db42
- compat-db43
- compat-libstdc++-296
- compat-libtermcap
- compat-openmpi
- compat-openmpi-psm
- compat-opensm-libs
- compiz
- compiz-gnome
- coreutils-libs
- cracklib-python
- cronie-noanacron
- ctan-cm-lgc-fonts-common
- ctan-cm-lgc-roman-fonts
- ctan-cm-lgc-sans-fonts

- ctan-cm-lgc-typewriter-fonts
- ctan-kerkis-fonts-common
- ctan-kerkis-sans-fonts
- ctan-kerkis-serif-fonts
- ctapi-common
- cvs-inetd
- c2050
- c2070
- dash
- dbus-c+
- dbus-qt
- devhelp
- dmz-cursor-themes
- dtach
- dvd+rw-tools
- eclipse-birt
- eclipse-callgraph
- eclipse-cdt
- eclipse-dtp
- eclipse-emf
- eclipse-gef
- eclipse-changelog
- eclipse-jdt
- eclipse-linuxprofilingframework
- eclipse-mylyn
- eclipse-mylyn-cdt
- eclipse-mylyn-java
- eclipse-mylyn-pde
- eclipse-mylyn-trac

- eclipse-mylyn-webtasks
- eclipse-mylyn-wikitext
- eclipse-nls
- eclipse-nls-ar
- eclipse-nls-bg
- eclipse-nls-ca
- eclipse-nls-cs
- eclipse-nls-da
- eclipse-nls-de
- eclipse-nls-el
- eclipse-nls-es
- eclipse-nls-et
- eclipse-nls-fa
- eclipse-nls-fi
- eclipse-nls-fr
- eclipse-nls-he
- eclipse-nls-hi
- eclipse-nls-hu
- eclipse-nls-id
- eclipse-nls-it
- eclipse-nls-ja
- eclipse-nls-ko
- eclipse-nls-ku
- eclipse-nls-mn
- eclipse-nls-nl
- eclipse-nls-no
- eclipse-nls-pl
- eclipse-nls-pt
- eclipse-nls-pt_BR

- eclipse-nls-ro
- eclipse-nls-ru
- eclipse-nls-sk
- eclipse-nls-sl
- eclipse-nls-sq
- eclipse-nls-sr
- eclipse-nls-sv
- eclipse-nls-tr
- eclipse-nls-uk
- eclipse-nls-zh
- eclipse-nls-zh_TW
- eclipse-oprofile
- eclipse-pde
- eclipse-platform
- eclipse-rcp
- eclipse-rpm-editor
- eclipse-rse
- eclipse-subclipse
- eclipse-subclipse-graph
- eclipse-svnkit
- eclipse-swt
- eclipse-valgrind
- ecryptfs-utils
- evolution-data-server-doc
- fakechroot
- fakechroot-libs
- fence-virt
- fence-virt-d-checkpoint
- file-devel

- firstaidkit
- firstaidkit-engine
- firstaidkit-gui
- foghorn
- fop
- gamin-devel
- gamin-python
- gconfmm26
- ggz-base-libs
- glade3
- gnome-disk-utility-libs
- gnome-disk-utility-ui-libs
- gnome-doc-utils
- gnome-doc-utils-stylesheets
- gnome-games
- gnome-media
- gnome-media-libs
- gnome-pilot
- gnome-pilot-conduits
- gnome-power-manager
- gnome-python2-bugbuddy
- gnome-python2-extras
- gnome-python2-gtkhtml2
- gnome-python2-libegg
- gnome-python2-libwnck
- gnome-python2-rsvg
- gnome-themes
- gnome-user-share
- gnome-vfs2-devel

- gnome-vfs2-smb
- graphviz-perl
- groff
- gsl-static
- gstreamer-python
- gthumb
- gtk+extra
- gtkhtml2
- gtksourceview2
- gtk2-engines
- guile
- gvfs-afc
- gvfs-archive
- hal-info
- hal-libs
- hal-storage-addon
- htdig
- hypervkvpd
- ibus-table-additional
- icedax
- icu4j-eclipse
- ipa-pki-ca-theme
- ipa-pki-common-theme
- ipw2100-firmware
- ipw2200-firmware
- jakarta-commons-discovery
- jakarta-commons-el
- jasper
- java_cup

- jdepend
- jetty-eclipse
- jsch
- jzlib
- kabi-yum-plugins
- kcoloredit
- kcoloredit-doc
- kdeadmin
- kdeartwork-screensavers
- kdebase-workspace-akonadi
- kdebase-workspace-python-applet
- kdegames
- kdegraphics
- kde-i18n-Arabic
- kde-i18n-Bengali
- kde-i18n-Brazil
- kde-i18n-British
- kde-i18n-Bulgarian
- kde-i18n-Catalan
- kde-i18n-Czech
- kde-i18n-Danish
- kde-i18n-Dutch
- kde-i18n-Estonian
- kde-i18n-Finnish
- kde-i18n-French
- kde-i18n-German
- kde-i18n-Greek
- kde-i18n-Hebrew
- kde-i18n-Hindi

- kde-i18n-Hungarian
- kde-i18n-Chinese
- kde-i18n-Chinese-Big5
- kde-i18n-Icelandic
- kde-i18n-Italian
- kde-i18n-Japanese
- kde-i18n-Korean
- kde-i18n-Lithuanian
- kde-i18n-Norwegian
- kde-i18n-Norwegian-Nynorsk
- kde-i18n-Polish
- kde-i18n-Portuguese
- kde-i18n-Punjabi
- kde-i18n-Romanian
- kde-i18n-Russian
- kde-i18n-Serbian
- kde-i18n-Slovak
- kde-i18n-Slovenian
- kde-i18n-Spanish
- kde-i18n-Swedish
- kde-i18n-Tamil
- kde-i18n-Turkish
- kde-i18n-Ukrainian
- kdelibs-apidocs
- kdelibs3
- kdelibs3-devel
- kde-l10n-Bengali-India
- kde-l10n-Frisian
- kde-l10n-Gujarati

- kde-l10n-Chhattisgarhi
- kde-l10n-Kannada
- kde-l10n-Kashubian
- kde-l10n-Kurdish
- kde-l10n-Macedonian
- kde-l10n-Maithili
- kde-l10n-Malayalam
- kde-l10n-Marathi
- kdemultimedia
- kdemultimedia-devel
- kdemultimedia-libs
- kdenetwork
- kdesdk
- kdesdk-libs
- kdeutils
- kdewebdev
- kdewebdev-libs
- kernel-debug
- kernel-debug-devel
- kernel-doc
- kiconedit
- kipi-plugins
- kipi-plugins-libs
- kmid
- kmid-common
- konq-plugins-doc
- krb5-appl
- kross-python
- ksig

- ksig-doc
- k3b
- k3b-common
- k3b-libs
- libao-devel
- libart_lgpl-devel
- libbonobo-devel
- libbonoboui-devel
- libburn
- libcroco-devel
- libdc1394
- libdiscid
- libesmtp-devel
- libexif-devel
- libgail-gnome
- libgcj
- libgcj-devel
- libgcj-src
- libglademm24
- libglade2-devel
- libgnomecanvas-devel
- libgnome-devel
- libgnomeui-devel
- libgphoto2-devel
- libgpod
- libgsf-devel
- libgxim
- libIDL-devel
- libidn-devel

- libisofs
- libitm
- libldb-devel
- libmatchbox
- libmtp
- libmusicbrainz
- libmusicbrainz3
- libnih
- liboil
- libopenraw-gnome
- libpanelappletmm
- libproxy-bin
- libproxy-python
- libreport-compat
- libreport-plugin-mailx
- libreport-plugin-reportuploader
- librtas (32 ビットのみ)
- libselinux-ruby
- libservicelog (32 ビットのみ)
- libsexy
- libtalloc-devel
- libtdb-devel
- libtevent-devel
- libtidy
- libvdp (32 ビットのみ)
- libwnck
- libXdmcp-devel
- log4cpp
- lpg-java-compat

- lucene
- lucene-contrib
- lx
- lynx
- MAKEDEV
- matchbox-window-manager
- mcstrans
- mesa-dri1-drivers
- min12xxw
- mod_auth_mysql
- mod_auth_pgsq1
- mod_authz_ldap
- mod_dnssd
- mrtg-libs
- mvapich-psm-static
- mx4j
- nspluginwrapper
- openct
- openhpi-subagent
- openssh-askpass
- ORBit2-devel
- osutil
- oxygen-cursor-themes
- PackageKit-yum-plugin
- paktype-fonts-common
- pam_passwdqc
- pbm2l2030
- pbm2l7k
- pcmciautils

- pcsc-lite-openct
- perl-BSD-Resource
- perl-Cache-Memcached
- perl-Class-MethodMaker
- perl-Config-General
- perl-Crypt-PasswdMD5
- perl-Frontier-RPC
- perl-Frontier-RPC-doc
- perl-Perlilog
- perl-String-CRC32
- perl-suidperl
- perl-Text-Iconv
- perl-Time-HiRes
- perl-YAML-Syck
- pessulus
- pilot-link
- pinentry-gtk
- piranha
- pki-symkey
- plpa-libs
- plymouth-gdm-hooks
- plymouth-theme-rings
- plymouth-utils
- policycoreutils-newrole
- policycoreutils-sandbox
- ppl
- prelink
- printer-filters
- psutils

- ptouch-driver
- pulseaudio-module-gconf
- pycairo-devel
- pygobject2-codegen
- pygobject2-devel
- pygobject2-doc
- pygtksourceview
- pygtk2-codegen
- pygtk2-devel
- pygtk2-doc
- pychart
- PyOpenGL [1]
- python-beaker
- python-Coherence
- python-crypto
- python-decoratortools
- python-enchant
- python-formencode
- python-fpconst
- python-genshi
- python-gtkextra
- python-cheetah
- python-ipaddr
- python-iwlib
- python-libguestfs [2]
- python-louie
- python-mako
- python-markdown
- python-markupsafe

- `python-matplotlib`
- `python-myghty`
- `python-paramiko`
- `python-paste`
- `python-paste-deploy`
- `python-paste-script`
- `python-peak-rules`
- `python-peak-util-addons`
- `python-peak-util-assembler`
- `python-peak-util-extremes`
- `python-peak-util-symbols`
- `python-prioritized-methods`
- `python-pygments`
- `python-pylons`
- `python-qpid`
- `python-qpid-qmf`
- `python-repoze-tm2`
- `python-repoze-what`
- `python-repoze-what-plugins-sql`
- `python-repoze-what-pylons`
- `python-repoze-what-quickstart`
- `python-repoze-who`
- `python-repoze-who-friendlyform`
- `python-repoze-who-plugins-sa`
- `python-repoze-who-testutil`
- `python-routes`
- `python-saslwrapper`
- `python-sexy`
- `python-sqlalchemy`

- python-tempita
- python-toscawidgets
- python-transaction
- python-turbojson
- python-tw-forms
- python-twisted
- python-twisted-conch
- python-twisted-core
- python-twisted-lore
- python-twisted-mail
- python-twisted-names
- python-twisted-news
- python-twisted-runner
- python-twisted-web
- python-twisted-words
- python-weberror
- python-webflash
- python-webhelpers
- python-webob
- python-webtest
- python-zope-filesystem
- python-zope-interface
- python-zope-sqlalchemy
- pywebkitgtk
- pyxf86config
- qpid-cpp-client
- qpid-cpp-client-ssl
- qpid-cpp-server
- qpid-cpp-server-ssl

- qpid-qmf
- qpid-tests
- qpid-tools
- qt-doc
- raptor
- rgmanager
- rome
- ruby-devel
- ruby-qpid
- ruby-qpid-qmf
- sabayon
- sabayon-apply
- sac
- samba-winbind-clients
- samba4
- samba4-client
- samba4-common
- samba4-dc
- samba4-dc-libs
- samba4-devel
- samba4-pidl
- samba4-swat
- samba4-test
- samba4-winbind
- samba4-winbind-clients
- samba4-winbind-krb5-locator
- saslwrapper
- sat4j
- saxon

- sblim-cmpi-dhcp
- sblim-cmpi-dns
- sblim-cmpi-samba
- sblim-tools-libra
- scenery-backgrounds
- seabios
- selinux-policy-minimum
- selinux-policy-mls
- setools-console
- sgabios-bin
- sigar
- sinjdoc
- smp_utils
- SOAPpy
- sound-juicer
- strigi-devel
- subscription-manager-migration-data
- subversion-javahl
- svnkit
- system-config-firewall
- system-config-firewall-tui
- system-config-network-tui
- system-config-services
- system-config-services-docs
- system-gnome-theme
- system-icon-theme
- taskjuggler
- tbird
- terminus-fonts

- tidy
- tigervnc-server
- tix
- tkinter
- trilead-ssh2
- tsclient
- tunctl
- TurboGears2
- unicap
- vorbis-tools
- wacomexpresskeys
- wdaemon
- webalizer
- webkitgtk
- ws-commons-util
- wsdl4j
- xfig-plain
- xfsprogs-devel
- xfsprogs-qa-devel
- xguest
- xmldb-api
- xmldb-api-sdk
- xmlgraphics-commons
- xorg-x11-apps
- xorg-x11-drv-acecad
- xorg-x11-drv-aiptek
- xorg-x11-drv-fpit
- xorg-x11-drv-hyperpen
- xorg-x11-drv-keyboard

- xorg-x11-drv-mouse
- xorg-x11-drv-mutouch
- xorg-x11-drv-openchrome
- xorg-x11-drv-penmount
- xorg-x11-server-Xephyr
- xsane
- xz-lzma-compat
- zd1211-firmware

3.5. 削除済みドライバー

以下のドライバーは、Red Hat Enterprise Linux バージョン 6 からバージョン 7 の間に削除され、現在はサポートされていません。

- 3c574_cs.ko
- 3c589_cs.ko
- 3c59x.ko
- 8390.ko
- acenic.ko
- amd8111e.ko
- avma1_cs-ko [3]
- avm_cs.ko
- axnet_cs.ko
- b1pcmpcia.ko
- bluecard_cs-ko
- bt3c_cs.ko
- btuart_cs.ko
- can-dev.ko
- cassini.ko
- cdc-phonet.ko
- cm4000_cs.ko
- cm4040_cs.ko

- cxgb.ko
- de2104x.ko
- de4x5.ko
- dl2k.ko
- dmfe.ko
- dtl1_cs.ko
- e100.ko
- elsa_cs.ko
- ems_pci.ko
- ems_usb.ko
- fealnx.ko
- fmvj18x_cs.ko
- forcedeth.ko
- ipwireless.ko
- ixgb.ko
- kvaser_pci.ko
- myri10ge.ko
- natsemi.ko
- ne2k-pci.ko
- niu.ko
- nmclan_cs.ko
- ns83820.ko
- parport_cs.ko
- pata_pcmcia.ko
- pcnet_cs.ko
- pcnet32.ko
- pppol2tp.ko
- r6040.ko
- s2io.ko

- sc92031.ko
- sdrichoh_cs.ko
- sedlbauer_cs.ko
- serial_cs.ko
- sis190.ko
- sis900.ko
- sja1000_platform.ko
- sja1000.ko
- smc91c92_cs.ko
- starfire.ko
- sundance.ko
- sungem_phy.ko
- sungem.ko
- sunhme.ko
- tehuti.ko
- teles_cs.ko
- tlan.ko
- tulip.ko
- typhoon.ko
- uli526x.ko
- vcan.ko
- via-rhine.ko
- via-velocity.ko
- vxge.ko
- winbond-840.ko
- xirc2ps_cs.ko
- xircom_cb.ko

3.6. 非推奨ドライバー

Red Hat Enterprise Linux 7 における非推奨のドライバーについては、[Red Hat カスタマーポータル](#) で提供されている最新版のリリースノートを参照してください。

[1] Red Hat Enterprise Linux 7.0 で削除され、Red Hat Enterprise Linux 7.1 で置き換えられました。Red Hat Enterprise Linux 7.3 でオプションチャンネルに追加されました。オプションチャンネルの詳細は [こちらのソリューション記事](#) を参照してください。

[2] Red Hat Enterprise Linux 7.0 でオプションリポジトリに移動し、Red Hat Enterprise Linux 7.1以降はベースチャンネルに戻りました。

[3] PCMCIA は、Red Hat Enterprise Linux 7 ではサポートされていません。USB などの新しいテクノロジーが代わりに使用されるようになりました。

第4章 移行関連の CUSTOMER PORTAL LABS

Red Hat Customer Portal Labs のツールは、パフォーマンスの向上、問題のトラブルシューティング、セキュリティ問題の特定、および設定の最適化を実行するユーザーをサポートするために設計されたツールです。この付録では、移行に関連する Red Hat Customer Portal Labs ツールの概要について説明します。Red Hat Customer Portal Labs はすべて <http://access.redhat.com/labs/> から入手できます。

Red Hat Enterprise Linux Upgrade Helper

[Red Hat Enterprise Linux Update Helper](#) は、バージョン 6.5、6.6、6.7、6.8 または 6.9 の Red Hat Enterprise Linux をバージョン 7.x にアップグレードするのに便利です。提供する必要がある情報は、アップグレードパスのみです。このアプリケーションでは、以下が表示されます。

- Red Hat Enterprise Linux をアップグレードする基本手順
- お客様のアップグレードシナリオに特化した既知の問題を回避する追加手順

このアプリケーションは以下のアップグレードパスをサポートします。

- 6.5 から 7.4
- 6.6 から 7.4
- 6.7 から 7.4
- 6.8 から 7.4
- 6.9 から 7.4

Product Life Cycle Checker

[Product Life Cycle Checker](#) は、リリース日、各フェーズの終了日、ライフサイクルの終了日など、Red Hat 製品のライフサイクル情報を確認するツールです。このツールでは、複数の製品を選択して、各製品の情報を一度に確認できます。