



Red Hat Enterprise Linux 6

移行計画ガイド

Red Hat Enterprise Linux 6 への移行

エディション 6

Red Hat Enterprise Linux 6 移行計画ガイド

Red Hat Enterprise Linux 6 への移行

エディション 6

Laura Bailey

Scott Radvan

法律上の通知

Copyright © 2013–2014 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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, 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 Software Collections 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 5 を稼働しているシステムの Red Hat Enterprise Linux 6 への移行を説明しています。

目次

第1章 はじめに	5
1.1. RED HAT ENTERPRISE LINUX 6	5
1.2. アプリケーションの互換性	6
第2章 インストール	8
2.1. カーネルと起動オプション	8
2.2. グラフィカルなインストーラー	8
2.2.1. デバイスとディスク	8
2.2.2. キックスタート	9
2.2.2.1. 動作変更	9
2.2.2.2. コマンドの変更	10
2.2.2.3. パッケージの変更	11
2.2.2.4. スクリプトの変更	12
2.2.2.5. 構文の変更	12
2.2.2.6. 違いに関する要約	12
2.2.2.7. pykickstart	13
2.2.3. ネットワーク設定	13
2.2.4. 製品のサブスクリプションとコンテンツの更新	14
2.3. テキストベースのインストーラ	15
第3章 ストレージとファイルシステム	16
3.1. KDUMP	16
3.2. RAID	16
3.2.1. アップグレード	16
3.3. EXT4	17
3.3.1. ext3 からの移行	17
3.3.2. 動作の変更	17
3.4. BLOCKDEV	17
3.5. テープデバイス	17
第4章 ネットワーク設定とサービス	18
4.1. インターフェースと設定	18
4.2. サービスの初期化	18
4.2.1. xinetd	18
4.2.2. ランレベル	18
4.2.3. Upstart	18
4.3. IP テーブル/ファイアウォール	19
4.4. APACHE HTTP SERVER	19
4.5. SAMBA	20
4.6. PHP	20
4.7. BIND	20
4.8. SNMP	21
4.9. NTP	22
4.10. PTP	22
4.11. KERBEROS	22
4.12. MAIL	23
4.12.1. Sendmail	23
4.12.2. Exim	23
4.12.3. Dovecot	23
4.12.3.1. Dovecot の設定	23
4.13. MYSQL®	24
4.13.1. DBD ドライバー	24

4.14. POSTGRESQL	24
4.14.1. データベースをアップグレードする	24
4.14.2. その他の変更	24
4.15. SQUID	24
4.16. BLUETOOTH	24
4.16.1. オンデマンドの Bluetooth サービス	24
4.17. CRON	24
4.17.1. Vixie cron と Cronie	24
4.18. ログ記録	25
第5章 コマンドラインツール	26
5.1. GREP	26
5.2. SED	26
5.3. PCRE	26
5.4. シェル	26
5.5. NAUTILUS	26
第6章 システム設定	28
6.1. ACPI	28
6.1.1. CPU のホットプラグ	28
6.2. ログ記録	28
第7章 デスクトップ	29
7.1. GDM の設定	29
第8章 セキュリティと認証	30
8.1. SELINUX	30
8.2. SSSD	30
8.3. LDAP	30
8.3.1. OpenLDAP	30
8.3.2. slapd 設定を変換する	30
8.4. チェックサム	31
8.5. PAM (プラグ可能な認証モジュール)	31
8.6. システムのユーザー	31
8.7. NIST SCAP 1.2 証明書	31
8.8. RSA および DSA キーの生成に関する変更点	32
第9章 システムのモニタリングとカーネル	33
9.1. DRACUT	33
9.2. IPVS (IP VIRTUAL SERVER)	33
9.3. ジョイスティックのサポート	33
9.4. ABRT (自動バグ報告ツール)	33
9.4.1. ABRT イベント	34
第10章 パッケージとドライバーに関する変更点	36
10.1. システム設定ツールの変更点	36
10.1.1. system-config-bind	36
10.1.2. system-config-boot	36
10.1.3. system-config-cluster	36
10.1.4. system-config-display	36
10.1.5. system-config-httpd	37
10.1.6. system-config-lvm	37
10.1.7. system-config-netboot	37
10.1.8. system-config-nfs	37
10.1.9. system-config-rootpassword	37

10.1.10. system-config-samba	37
10.1.11. system-config-securitylevel	37
10.1.12. system-config-soundcard	37
10.1.13. system-config-switchmail	37
10.1.14. Preupgrade Assistant	38
10.1.15. Red Hat Upgrade Tool	38
10.2. BASH (BOURNE-AGAIN SHELL)	38
10.2.1. 正規表現	39
10.3. その他のパッケージ変更点	40
10.3.1. 新しいパッケージ	40
10.3.2. 更新パッケージ	41
10.3.3. 廃止パッケージ	42
10.3.4. 非推奨パッケージ	43
10.4. ドライバー関連の変更点	44
10.4.1. 廃止ドライバー	44
10.4.2. 非推奨ドライバー	45
10.4.3. 廃止カーネルコンポーネント	45
10.5. ライブラリ関連の変更点	45
付録A 改訂履歴	46

第1章 はじめに

移行計画ガイドでは Red Hat Enterprise Linux 5 インストールのマイナーバージョンを Red Hat Enterprise Linux 6 に移行する際に注目すべき主な動作変更を中心とした移行に関する説明を記載しています。

本ガイドでは Red Hat Enterprise Linux 5 と Red Hat Enterprise Linux 6 の製品上の変更点に関するガイドラインを説明することで Red Hat Enterprise Linux 6 の使い勝手を向上させることを目的としています。ただし、本ガイドは新機能をすべて説明するためのガイドではありません。Red Hat Enterprise Linux 5 の一部であり Red Hat Enterprise Linux 6 では変更されているアプリケーションやコンポーネントの動作、またはその機能が別のパッケージによって置換されているパッケージに焦点を絞って説明しています。

1.1. RED HAT ENTERPRISE LINUX 6

Red Hat Enterprise Linux はオープンソースコンピューティング業界で主要となるプラットフォームです。サブスクリプション方式による販売でその真価を継続的に配信、トップクラス企業のハードウェアベンダーやソフトウェアベンダーによる認定を受けています。デスクトップからデータセンターまで、Red Hat Enterprise Linux はオープンソース技術による革新と真のエンタープライズクラスプラットフォームとしての安定性との統合を実現しています。

Red Hat Enterprise Linux 6 は、ミッションクリティカルな企業用コンピューティング向けに設計された次世代オペレーティングシステムの総合パッケージになります。本リリースは以下のアーキテクチャで単独キットとして使用できます。

- i386
- AMD64/Intel64
- System z
- IBM Power (64-bit)

本リリースではサーバーやデスクトップ全体における改善、オープンソース開発で得た技術の集大成を提供しています。以下に、数多くの改善点の一部、また本リリースに収納されている新機能などを示します。

電力管理

Tuned による適応型システムチューニング、電力管理 (ASPM、ALPM)、PowerTOP による電力消費測定、ウェイクアップ低減のためのアプリケーションスタックに対する各種改善点、ティックレスカーネルなど

次世代のネットワーキング

IPv6 の総合サポート (NFS 4、CIFS、モバイルサポート [RFC 3775]、ISATAP サポート)、FCoE、iSCSI、新たに改善が加えられた mac80211 ワイヤレススタックなど

信頼性、可用性、サービスの有用性

業界間の連携により大部分のハードウェアに RAS 機能および NUMA アーキテクチャを搭載させることでシステムレベルでの強化を実現

詳細レベルの制御と管理

CFS (Completely Fair Scheduler) と CG (Control Groups) によるカーネル内でのリソース管理やスケジューラの改善

スケーラブルなファイルシステム

デフォルトのファイルシステムは **ext4**、パフォーマンス性の高い堅固でスケーラブルなファイルシステムは **xfs** で提供

仮想化

KVM に対するパフォーマンス関連の改善および新機能の追加 (sVirt ではホスト、仮想マシンおよびデータをゲストによる侵害から保護、SRIOV および NPIV では物理デバイスの仮想使用におけるパフォーマンスを向上、libvirt ではカーネルの CG コントローラー機能の活用)

企業レベルのセキュリティ強化

SELinux では、使い易さを向上させアプリケーションのサンドボックス機能やシステムサービスの対象範囲を大幅に拡張させている一方、SSSD では識別や認証サービスに対する統合アクセスやオフライン使用向けのキャッシング機能なども提供

開発およびランタイムのサポート

SystemTap (再コンパイルすることなく実行中のカーネルを操作)、ABRT (バグ情報のシンプルな収集)、および GCC (バージョン 4.4.3)、glibc (バージョン 2.11.1)、GDB (バージョン 7.0.1) に対する各種の改善

1.2. アプリケーションの互換性

Red Hat Enterprise Linux の本リリースでは、旧バージョンでの実行を目的として設計されているアプリケーションを最小限の調整で引き続き実行することができるよう依存性を提供しています。このため、本リリースと旧バージョン間で変更された可能性のあるレガシーなインターフェースを維持するために重要となるライブラリーの旧バージョンが収納されています。これらのライブラリーは C/C++ で記述されたアプリケーションの依存関係を解決することが主な目的です。

マイナーリリースが異なる場合はアプリケーションの再テストや再認定を行う必要はありません。Red Hat Enterprise Linux リリースの任意のバージョンで実行しているアプリケーションはそのリリースの耐用年数の期間中は継続的に実行できることを Red Hat Enterprise Linux の互換性ポリシーで保証しています。例えば、Red Hat Enterprise Linux 6.0 で認定されたアプリケーションは Red Hat Enterprise Linux 6.1 などのすべてのマイナーリリースでも完全互換となります。

互換性パッケージの詳細は以下の表を参照してください。

表1.1 互換性ライブラリー

パッケージ	説明
compat-db	Berkeley DB データベース互換性ライブラリーです。Berkeley Database (Berkeley DB) はプログラム型のツールキットで従来のアプリケーションおよびクライアント/サーバー型アプリケーションのいずれに対しても組込み型のデータベースサポートを提供します。このパッケージには、旧リリースに収納されていた Berkeley DB の各種バージョンが含まれています。
compat-expat1	Expat はストリーム指向の XML 構文解析ツールです。このパッケージは旧バージョンとのライブラリー互換性を提供します。

パッケージ	説明
compat-glibc	glibc はシステムコールやその他の基本的な機能に使用される C ライブラリーです。旧式の glibc バージョンを必要とするバイナリーのコンパイルに互換性 (およびランタイムライブラリー) を提供し、Red Hat Enterprise Linux の本リリースで実行できるようにします。
compat-libf2c-34	このパッケージは、Fortran 77 共有ライブラリーの旧バージョンを提供します。動的にリンクされた Fortran 77 プログラムの実行に必要となります。
compat-libgcc-296	GCC の旧バージョンとの互換性を維持するため 2.96 libgcc.a ライブラリーとサポートオブジェクトファイルが収納されています。
compat-libgfortran-41	このパッケージには GCC 4.1.x でコンパイルした Fortran アプリケーションとの互換性に要される Fortran 95 ランタイムライブラリーが含まれています。
compat-libstdc++-295	GNU 標準 C++ ライブラリーのバージョン 2.95 との互換性を提供します。
compat-libstdc++-296	GNU 標準 C++ ライブラリーのバージョン 2.96 との互換性を提供します。
compat-libstdc++-33	GNU 標準 C++ ライブラリーのバージョン 3.3 との互換性を提供します。
compat-libtermcap	このパッケージは termcap ベースの旧式プログラム用の互換性を提供します。
compat-openldap	OpenLDAP は LDAP (Lightweight Directory Access Protocol) アプリケーションおよび開発ツールセットのオープンソース版です。compat-openldap パッケージには OpenLDAP 共有ライブラリーの旧バージョンが含まれています。このパッケージは一部のアプリケーションで時々必要とされることがあります。
openssl098e	OpenSSL 0.98e を提供します。一部の SSL アプリケーションで必要とされることがあります。

第2章 インストール

本セクションでは、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 5 とのインストール手順の違いについて簡単に説明します。使用されている Red Hat Enterprise Linux 5 のリリースによってはすでに導入されている場合もあるため、ここで説明しているオプションや技術がすべて適しているとは限りません。

2.1. カーネルと起動オプション

- Red Hat Enterprise Linux をインストールする前に、**boot:** プロンプトで **memtest86** と入力するとメモリーのテストを行うことができます。このオプションにより **Anaconda** システムインストーラではなく、独立型のメモリーテストソフトウェア **Memtest86** が実行されます。このソフトウェアを一度実行すると、**Esc** キーを押すまで **Memtest86** メモリーテストは繰り返して実行が継続されます。
- モジュールの読み込み順序を指定する際、旧式の **scsi_hostadapter** オプションの代わりに **rdloaddriver** カーネルパラメーターが今後は必要になります。
- **KMS (Kernel Modesetting)** はグラフィックモードを初期化する役割をカーネルに渡す機能です。デフォルト有効になります。KMS では次を提供します。
 - グラフィカル起動の改善
 - ファストユーザースイッチの高速化
 - シームレスな X サーバー切り替え
 - グラフィカルなパニックメッセージ

システムの起動時に **nomodeset** を **boot:** 行に付けるとすべてのドライバーに対して KMS を無効にすることができます。

2.2. グラフィカルなインストーラー

本セクションでは、グラフィカルなインストーラーで変更になっている動作を説明します。

2.2.1. デバイスとディスク

- **/dev/hdX** というようなデバイス名の使い方は IDE ドライブ用の **i386** と **x86_64** のアーキテクチャでは非推奨になり、**/dev/sdX** という使い方に変更になります。この変更は **PPC** アーキテクチャには適用されません。
- インストールで **Smart Array** カードが検出されない場合はインストーラーのプロンプトで **linux isa** と入力します。これで必要なカードを手作業で選択できるようになります。
- 旧式の IDE ドライバーはデバイス毎に最大 **63** パーティションまで対応していたのに対し、**SCSI** デバイスはデバイス毎に **15** パーティションに制限されています。**Anaconda** は Red Hat Enterprise Linux の他の部分と同じように新しい **libata** ドライバーを使用しますので、インストール時やアップグレード中に **15** を越えるパーティションは検出できません。**15** を越えるパーティションで構成されるシステムをアップグレードしている場合は、ディスクを **LVM (論理ボリュームマネージャー)** に移行することをお勧めします。
- カーネルによるストレージデバイスの処理方法が変更されたことで **/dev/hdX** や **/dev/sdX** などのデバイス名が旧リリースで使用されていた値と一致しないことがあります。**Anaconda** は

パーティションラベルに依存することでこの問題を解決します。このパーティションラベルがない場合、**Anaconda** はパーティションにラベル付けが必要なことを警告してきます。**LVM** (論理ボリューム管理) とデバイスマッパーを使用するシステムの場合には、通常は再ラベルの必要はありません。

- **LUKS (Linux Unified Key Setup)** 使用が同梱され、**root** ファイルシステムなど、暗号化されたブロックデバイスへのインストールにも対応するようになります。**LUKS** については『**Red Hat Enterprise Linux インストールガイド**』を参照してください。
- 全ての **IDE RAID** コントローラーに対応しているとは限りません。**dmraid** がご使用の **RAID** コントローラーにまだ対応していない場合は、**Linux** ソフトウェア **RAID** を設定して複数のドライブ群を **RAID** アレイに結合させることが可能です。対応しているコントローラーの場合はコンピュータの **BIOS** 内で **RAID** 機能を設定してください。
- **Red Hat Enterprise Linux 6** に収納されている **GRUB** のバージョンは **ext4** に対応するようになるため、**Anaconda** では **/boot** や **root** のパーティションも含めいずれのパーティションでも **ext4** ファイルシステムを使用できるようになります。

2.2.2. キックスタート

本セクションでは、自動化したインストール(キックスタート)での動作変更について説明します。

2.2.2.1. 動作変更

- 以前は、キックスタートファイルに **network** の行を含ませないことで「ネットワークの設定に **DHCP** を使用する」という指示になることを意図する一方、キックスタートの他の部分では「行が含まれていない場合、インストールを中断して入力を求める」という指示になるため、矛盾が生じていました。今後は **network** の行を含ませない場合、ネットワークアクセスが必要になった際にインストールを中断して入力を求めるという指示になります。インストールを中断させることなく **DHCP** を使用したい場合はキックスタートファイルに **network --bootproto=dhcp** を追加してください。また、**--bootproto=query** は非推奨のオプションになります。インストールの初期段階でネットワーク設定の入力を求めるプロンプトを表示させたい場合は **asknetwork** オプションを使用してください。
- **Red Hat Enterprise Linux** の旧バージョンでは、値を付けずに **ks** オプションをシステムに渡す場合、**next-server** **DHCP** オプションを使ってキックスタートファイルを持たせている **NFS** サーバーを指定していました。**Red Hat Enterprise Linux 6** ではこの **DHCP** オプションが **server-name** に変更になっています。
- 従来、ディスクの参照はキックスタートの全過程でデバイスノード名 (**sda** など) が使用されていました。**Linux** カーネルがより動的なメソッドに転換していくにつれ、再起動後のデバイス名が必ずしも一致しなくなっています。このため、キックスタートスクリプトでデバイス名を使用するのが難しくなっています。一貫したデバイス名を維持するには、デバイスノード名の代わりに **/dev/disk** 配下のアイテムを使用します。例えば、以下のデバイスノード名を使用していたとします。

```
part / --fstype=ext4 --onpart=sda1
```

代わりに、以下のいずれかのようにすることができます。

```
part / --fstype=ext4 --onpart=/dev/disk/by-path/pci-0000:00:05.0-
scsi-0:0:0:0-part1
part / --fstype=ext4 --onpart=/dev/disk/by-id/ata-
ST3160815AS_6RA0C882-part1
```

これにより、一貫性を欠く **sda** ではなく、再起動後も不変となるディスク参照を行わせることができるようになります。特に大規模なストレージ環境で役に立ちます。

- また、複数ディスクの参照にシェル系のエントリを使用することもできます。主に大規模なストレージ環境で **clearpart** と **ignoredisk** コマンドを使い易くするのが目的です。例えば、以下のデバイスノード名を使用していたとします。

```
ignoredisk --drives=sdaa, sdab, sdac
```

代わりに、以下のようにすることができます。

```
ignoredisk --drives=/dev/disk/by-path/pci-0000:00:05.0-scsi-*
```

- 以前のバージョンに比べ、キックスタートはエラーで中断されることが多くなります。例えば、存在しないディスクを参照するとインストールが中断されエラーを通知します。深刻な問題を招く前にキックスタートファイル内のエラーを検出するよう設計されているためです。副作用として、各種のマシン構成に対して汎用となるよう設計されているファイルの場合、エラーの発生頻度が高くなる場合があります。これらについてはその都度、対処してください。
- キックスタートのネットワーク情報に使用していた **/tmp/netinfo** ファイルが削除されています。**Anaconda** ではインターフェースの設定にはデフォルトで **NetworkManager** を使用できるようになります。また、設定は **/etc/sysconfig/network-scripts/** 内の **ifcfg** ファイルに保存されるようになります。この新しい場所は **%pre** スクリプトおよび **%post** スクリプト用ネットワーク設定のソースとして使用することができます。

2.2.2.2. コマンドの変更

本セクションでは、コマンドとそのオプションに対する最も重要な変更について説明していきます。

- **network --device** オプションでは、デバイス名ではなく **MAC** アドレスでデバイスを参照できるようになります。ディスクと同様に、ネットワークデバイス名もデバイスが検出される順番に応じて再起動後に変化します。キックスタートで一貫した命名を行うためには、以下のようなエントリを使用することができます。

```
network --device=00:11:22:33:44:55 --bootproto=dhcp
```

- **langsupport**、**key**、**mouse** の各コマンドは削除されています。これらのコマンドを使用すると構文エラーになります。**monitor** コマンドも非推奨になっています。

langsupport の代替として、キックスタートファイルの **%packages** セクションに該当するグループを追加します。例えば、フランス語のサポートを含める場合は **@french-support** を追加します。

インストール時にインストールキーの入力は求められなくなるため、**key** オプションの代替はありません。ご使用ファイルからこのオプションを削除しておいてください。

マウスおよびモニターは **X.Org** で自動検出、設定ができるため、**mouse** と **monitor** のコマンドは必要なくなります。これと同じ理由で **xconfig --resolution=** コマンドも無効になります。これらのコマンドはすべてファイルから削除して構いません。

- **part --start** と **part --end** のコマンドは非推奨となっているため何の効果もありません。**Anaconda** では特定セクターの境界にはパーティションを作成できなくなります。より厳密なレベルのパーティション設定を必要とする場合は、**%pre** に外部のツールを使用し、

Anacondaに **part --onpart** コマンドを使って既存のパーティションを使用するよう指示します。これ以外は、サイズを特定したパーティションの作成または **--grow** を使ったパーティション作成になります。

- **%post** で手作業によりグループ群を作成する代わりに、**group** コマンドを使用してグループ群を作成することができるようになります。詳細についてはキックスターートのドキュメントを参照してください。
- **rescue** コマンドを使用すると自動的に復元や修復を行うインストーラーのレスキューモードに入ります。**rescue** コマンドには **--nomount** (ファイルシステムをマウントしない) オプションや **--romount** (読み取り専用モードでマウントする) オプションを使用することもできます。
- **sshpw** コマンドが追加されています。このコマンドはインストール実行中にリモートによるログインが行われる環境で作成されるアカウントの制御に使用します。
- **updates** コマンドが追加され、これを使用するとインストール中に**updates.img** ファイルの場所を指定することができるようになります。
- **fcoe** コマンドを使用するとインストーラーが指定ネットワークインターフェースに接続している FCoE の場所をアクティブにすることができるようになります。
- デフォルトの **autopart** アルゴリズムが変更されています。すべてのマシンに対して **autopart** は **/boot** (またはアーキテクチャで要される他の特殊なブートローダパーティション) と **swap** を作成します。最低でも **50 GB** の空きディスク領域を持つマシンには、**autopart** は適当なサイズの **root** パーティション (**/**) を作成してから残りを **/home** に割り当てます。空き領域の少ないマシンの場合、**root** (**/**) のみを作成します。

/home ボリュームを作成したくない場合は、**autopart** は使用しないでください。代わりに、**/boot**、**swap**、及び **/** を指定し、必要に応じて **root** ボリュームが拡張できるようにします。

- **Anaconda** はインストール中に視認可能なデバイスを制御できる新しいストレージフィルタリングインターフェースを収納するようになります。このインターフェースは既存の **ignoredisk**、**clearpart**、**zerombr** のコマンドに相当します。**ignoredisk** はオプションのため、これをキックスタートファイルから除外しても、インストール中にフィルタのユーザーインターフェースは表示されません。このインターフェースを使用したい場合は以下を追加します。

```
ignoredisk --interactive
```

- **/tmp/partition-include** ファイルの **--size=1 --grow** オプションは使用できなくなります。適当なデフォルトサイズを指定しなければなりません。パーティションはそれに応じて拡張します。

2.2.2.3. パッケージの変更

これらの変更は **%packages** セクションに影響します。

- **--ignoreDeps** と **--resolveDeps** の引数は削除されています。**Anaconda** は自動的に依存関係を解決します。ただし、依存関係を満たさないパッケージのインストールは省略されません。
- すべてデフォルトの設定でインストールを行うデフォルトの **GUI** インストールと全く同じパッケージセットをキックスタートでインストールさせたい場合は以下を追加します。

```
%packages --default
```

```
%end
```

- また、オプションとして複数アーキテクチャのインストール用にインストールしたいパッケージのアーキテクチャを指定することもできます。例えば、

```
%packages  
glibc.i686  
%end
```

これによりパッケージセットに x86 の `glibc` パッケージが追加されることになります。互換上 x86 のパッケージを必要とする x86-64 システムで役に立ちます。

- `%packages` セクションの全てのパッケージとグループを監査して移行することはできません。一部のパッケージやグループは削除されていたり、追加されていたり、また名前がへんこうされているものもあります。詳細についてはリリースノートを参照してください。

2.2.2.4. スクリプトの変更

これらの変更は `%pre`、`%post`、及び `%traceback` のスクリプトの使用に影響します。

- スクリプトの実行中に行われるエラーのログ記録が改善されています。スクリプトがその実行後に削除されなくなるため、スクリプトの確認が可能になります。何が実行されたのかを確認することができるため、スクリプトが動的に生成されるようなシステムでもっとも役に立ちます。また、全スクリプトの `stderr` と `stdout` の出力が常にログ記録されます。ただし重要な副作用がひとつあります。スクリプトが対話式のプログラムを使用する場合、そのスクリプトのヘッダに `--logfile=/dev/tty3` を渡す必要があり、これを行わないとプログラムと対話することができません。

2.2.2.5. 構文の変更

中核となるキックスタート構文への変更は非常に稀ですが、知っておくべき重要な構文の変更が 2 つあります。

- `%include` オプションはファイル名の他にも URL を引数として受け取ることができるようになります。
- `%packages`、`%post`、`%pre`、`%traceback` などのセクションには `%end` セクションは必要ありません。`%end` セクションが含まれていないと非推奨の警告が表示されますがスクリプトはそのまま続行されます。

2.2.2.6. 違いに関する要約

Red Hat Enterprise Linux 6 でのコマンドとオプションの違いを一覧形式で説明します。

削除されたコマンド:

- `key`
- `langsupport`
- `mouse`

非推奨のコマンド:

- `monitor`

- `xconfig --resolution`

追加されたコマンド:

- `fcoe`
- `group`
- `rescue`
- `sshpw`
- `updates`

2.2.2.7. pykickstart

`pykickstart` パッケージには、移行を容易にすることができるユーティリティが含まれています。最新のパッケージがインストールされていることを確認してください。`ksverdiff` コマンドは開始と終了の構文バージョンを受け取り、2種類のバージョンで異なるコマンドとオプションを報告します。

「新規」、「非推奨」、「削除済み」のコマンドとオプションを示します。例えば、

```
$ ksverdiff --from RHEL5 --to RHEL6

The following commands were removed in RHEL6:
langsupport mouse key

The following commands were deprecated in RHEL6:
monitor

The following commands were added in RHEL6:
sshpw group rescue updates fcoe
...
```

`ksvalidator` コマンドを使用すると、キックスタートファイルの有効性をチェックすることもできます。このコマンドはファイルの有効性を指定したキックスタート構文のバージョンに対してチェックします。ただし、`part --ondisk=sdr` を指定したがそのようなデバイスは存在しないなど、インストール時にしか発生しない問題については通知できません。使用例を示します。

```
$ ksvalidator --version RHEL6 my-rhel5-ks.cfg
```

2.2.3. ネットワーク設定

本セクションでは、ネットワーク設定に関連したグラフィカルインストーラの動作変更について説明します。

- **Anaconda** はインストール中のネットワークインターフェースの設定に **NetworkManager** を使用するようになります。**Anaconda** の主要なネットワークインターフェース設定の画面は削除されています。インストール中に詳細が必要な場合にのみ、ネットワーク設定の詳細入力が必要と求められます。インストール中に使用した設定は、後日、使用できるようシステムに書き込まれます。
- PXE ブートの時にインストールメディア用に NFS 経由でマウントした `.iso` ファイルを使用する場合は、コマンドラインに `repo=nfs:server:/path/` を追加します。`install.img` と `product.img` ファイルの抽出、`nfs:server:/path/images/` ディレクトリへの配置も

必要になります。 `product.img` ファイルには他の定義や各種インストークラスが含まれています。

- 複数のネットワークインターフェースを持つシステムの中には、システムの BIOS で認識されている最初のネットワークインターフェースに `eth0` を割り当てないものがあります。これにより、インストーラーが本来 PXE で使用されていたものと異なるネットワークインターフェースを使用しようとする場合があります。この動作を変更するには、`pxelinux.cfg/*` 設定ファイル内で以下を使用してください。

```
IPAPPEND 2
APPEND ksdevice=bootif
```

この設定オプションで、システム BIOS と PXE が使用するのと同じネットワークインターフェースをインストーラーに使用させるようにします。以下のオプションを使用することで、ネットワークスイッチにリンクされていることを検出する最初のネットワークデバイスをインストーラーに使用させるようにします。

```
ksdevice=link
```

2.2.4. 製品のサブスクリプションとコンテンツの更新

Red Hat Enterprise Linux 6 ではコンテンツの配信およびサブスクリプション管理に柔軟で最新のサービスを導入しています。本セクションではコンテンツサービスに対する変更について説明します。

- **Red Hat Network** ホストの環境はチャンネルベースのサブスクリプションから製品と品質ベースのサブスクリプション使用に更新されます。新しい証明書ベースの **RHN** では、サブスクリプションとシステムの管理用のクライアントツール群が再設計されているため、新しいサブスクリプションとコンテンツ配信ネットワーク (CDN) で動作します。

従来のチャンネルベースの **RHN** も **RHN Classic** で利用できます。

これらの 2 種類のサブスクリプションサービスは同じプラットフォームで利用可能です。並列テクノロジーですべてのサブスクリプションをいずれの方法でも登録、管理することができます。

Satellite または **Proxy** サーバーを使用している環境の場合、従来のチャンネルベースのサブスクリプションシステムが継続して使用されるため、システムの登録は **RHN Classic** で行われることになります。

- 新しいコンテンツサーバーのオプション、**Red Hat Network Classic** が `firstboot` ウィザードに追加されています。これは更新された **RHN** および **CDN** ではなく従来のチャンネルベースの **RHN** を使用します。デフォルトの **Red Hat Network** オプションは証明書ベースの新しい **Red Hat Network** 管理プラットフォームを使用します。
- 証明書ベースの **RHN** と **RHN Classic** は相互運用が可能です。システムを一方のサービスで登録する場合、他方のサービスはそれを認識できるため警告を発生することはありません。システムの登録はいずれか一方のサービスでのみ行ってください。両方のサービスで登録を行うことはできません。

現在、**RHN Classic** を使用しているシステムを新しい証明書ベースの **Red Hat Network** を使用するシステムに直接的に移行させるパスはありません。システムを一方のサービスから他方に移動する場合は 2 通りの方法があります。

- `yum` ではなく起動 ISO を使ってシステムを Red Hat Enterprise Linux 6.1 またはそれ以降に更新します。

- 手作業で RHN Classic からシステムを削除しホストの記録を消してから、そのシステムを証明書ベースの Red Hat Network に Red Hat サブスクリプションマネージャツールを使って登録します。
- クライアントツールの新しいセットとなる Red Hat サブスクリプションマネージャ GUI と CLI は証明書ベースの RHN からサブスクリプションを管理できるよう Red Hat Enterprise Linux 6.1 およびそれ以降で提供されます。RHN Classic で管理しているシステムを操作できるよう既存の `rhn_*` ツールも使用できます。

2.3. テキストベースのインストーラ

Red Hat Enterprise Linux 6 のテキストモードのインストールオプションは、以前のバージョンに比べてかなり合理化されています。テキストモードでのインストールでは、以前プロセスの一部であった複雑な手順が省略され、簡単でわかりやすいインストールが行えるようになります。本セクションでは、テキストベースのインストーラを使用する際の動作変更について説明しています。

- **Anaconda** は、ベースとコアのグループからのみ自動的にパッケージを選択します。インストールプロセスが終了した時点でシステムが正しく動作し、更新や新しいパッケージのインストール準備が整った状態にするにはこれらのパッケージで十分です。
- システムのどこに Red Hat Enterprise Linux をインストールすべきかを指定できるよう **anaconda** によって以前のバージョンからの初期画面がまだ表示されます。ドライブ全体の使用、既存の Linux パーティションの削除、またはドライブ上の空き領域の使用のいずれかを選択できます。ただし、**anaconda** はパーティションのレイアウトを自動的にセットするため、この基本レイアウトからのパーティションやファイルシステムの追加または削除に関してユーザーに入力は求めなくなります。インストール時にカスタムなレイアウトを必要とする場合は、VNC 接続かキックスタートインストールでグラフィカルなインストールを実行する必要があります。LVM (論理ボリューム管理)、ファイルシステムの暗号化、サイズ変更が可能なファイルシステムなど高度なオプションについてはグラフィカルモードかキックスタートでしか利用できません。グラフィカル (VNC) インストールについては『Red Hat Enterprise Linux インストールガイド』を参照してください。
- **Anaconda** はテキストベースのインストーラで自動的にブートローダの設定を行うようになります。
- キックスタートを使用したテキストモードのインストールは旧バージョンと同じ方法で実行されます。ただし、パッケージの選択、高度なパーティション設定、ブートローダ設定についてはテキストモードでは自動化されるようになるため、**anaconda** はこれらの手順で必要とされる情報についてユーザーによる入力を求めることができなくなります。したがって、キックスタートファイルにはパッケージ選択、パーティションの設定、およびブートローダの設定情報を必ず含ませるようにしてください。いずれかの情報が見つからないと **anaconda** はエラーメッセージを出力して終了することになります。

第3章 ストレージとファイルシステム

3.1. KDUMP

`kdump` を行う際、ダンプ先に `ext4`、`XFS`、`btrfs` などのファイルシステムを使用することができるようになります。ファイルシステムの詳細については Red Hat Enterprise Linux 6 の『ストレージ管理ガイド』を参照してください。

3.2. RAID

3.2.1. アップグレード

`dmraid` セットから `mdraid` セットへのアップグレード操作はサポートされていません。この種類のアップグレードが試行されると警告が表示されます。既存の `mdraid` セットからのアップグレード、および新規の `mdraid` セットの作成は可能です。

Raid セットをアップグレードすると、新しいデフォルトのスーパーブロックが問題を引き起こす恐れがあります。この新規のスーパーブロック形式 (`RAID1 /boot` パーティションの作成時を除き全デバイスで使用される) はアレイの先頭に位置するようになるため、ファイルシステムや LVM データはすべてそのパーティションの先頭より補正された位置になります。アレイを稼働していないと、LVM およびファイルシステムの `mount` コマンドは有効なボリュームやファイルシステムデータを持つデバイスを検出できない場合があります。これは意図した動作であり、`RAID1` アレイ内に単独のディスクをマウントしたい場合には、その単独ディスクしか持っていないアレイを起動してからそのアレイをマウントする必要があるということであり、ベアディスクを直接マウントすることはできません。この変更が行われたのは、再同期が強制されていない場合に、ベアディスクを直接マウントすると警告無しにアレイを破損させる可能性があるためです。

次の再起動で、`RAID` システムはアレイ内に含まれていなかったディスクは互換性がないと判定する可能性があります。そのデバイスをアレイから切断します。これも正常な動作となります。アレイに他のディスクを再追加する準備ができたなら、`mdadm` コマンドを使用してこのディスクをアレイにホット追加 (動作中の追加) します。この時点でディスクの変更部分 (書き込み予定のビットマップがある場合) またはディスク全体 (ビットマップがない場合) の再同期が行われます。これでアレイは再度同期化されることになります。この時点以降は、アレイは正常に構成されたと判定されるためデバイスがアレイから切断されることはなくなります。

新規のスーパーブロックは `mdraid` アレイと言う名前の概念をサポートします。アレイ同士を区別するための旧式のアレイ一覧表記 (例、`/dev/md0`、`/dev/md1` など) への依存は無くなりました。アレイには任意の名前 (`home`、`data`、`opt` など) を選択できるようになります。 `--name=opt` オプションを使用して選択した名前で作成します。どのような名前をアレイに付けてもその名前は `/dev/md/` 内に作成されることになります。(ただし、名前として完全パスを与えた場合はそのパスが作成されます。また、「0」などの数字をひとつだけ与えると `mdadm` は旧式の `/dev/mdx` スキームを使ってアレイを開始します。) `Anaconda` インストーラでは現時点ではアレイ名の選択はできません。代わりに過去にアレイがどのように作成されたかをエミュレートする方法としてシンプルな番号形式を使用します。

この新しい `mdraid` アレイは書き込み予定ビットマップの使用に対応しています。これによりシステムはアレイから問題のある部分を識別することができるようになるため、不正なシャットダウンが発生した場合にはその問題となる部分のみが再同期を必要とし、ディスク全体を再同期する必要がなくなります。このため再同期に必要な時間が劇的に減少します。新たに作成したアレイには書き込み予定ビットマップが必要に応じて自動的に追加されます。例えば、`swap` に使用されるアレイおよび非常に小規模のアレイ (`/boot` アレイなど) の場合は書き込み予定ビットマップを持たせてもその利点ははありません。デバイス上で `mdadm --grow` コマンドを使用してアップグレードを完了した後以前から既存のアレイに書き込み予定ビットマップを追加することはできますが、書き込み予定ビットマップによ

り若干のパフォーマンス劣化が発生します (チャンクサイズ **65536** のビットマップで約 **3%** から **5%** のダウンですが、**8192** 位の小さいサイズのビットマップチャンクの場合は **10%** またはそれ以上になることがあります)。アレイに書き込み予定ビットマップを追加する場合には適度に大きめのチャンクサイズを維持するのが最適ということになります。推奨サイズは **65536** です。

3.3. EXT4

3.3.1. ext3 からの移行

ext4 に移行する場合は ext4 ファイルシステムを新規にフォーマットする必要があります。ext3 から ext4 への移行はサポートされていません。現在パーティションにあるデータには ext4 の利点や変更が活用されないため ext4 で提供している多くの利点は利用できません。

正しく新規フォーマットされている ext4 に移行ができない場合、Red Hat では既存の ext3 ファイルシステムを引き続き使用されることをお勧めします。

3.3.2. 動作の変更

Red Hat Enterprise Linux 6 では ext4 に完全対応ようになるため、新規のインストールに対しては ext4 がデフォルトのファイルシステムとなります。本セクションでは、この新しいファイルシステムにより導入される主な動作変更について説明しています。

- 収納されている GRUB ブートローダのバージョンは ext4 パーティションに対して完全対応となります。また、インストーラを使用する際も ext4 パーティション上にあらゆる **/boot** ファイルシステムを配置することができます。
- 収納されている e2fsprogs パッケージのバージョンは ext4 と完全互換になります。
- e4fsprogs パッケージを同梱した Red Hat Enterprise Linux 5.3 では、ext4 ファイルシステムを作成すると **ext4dev** ファイルシステムタイプになることがありました。これらのファイルシステムを開発バージョンとして識別する **test_fs** 機能フラグは、コマンド **tune2fs -E ^test_fs** を使用して削除することができます。これを行うことにより、ファイルシステムが通常の ext4 ファイルシステムとして認識されるようになります。

3.4. BLOCKDEV

blockdev --rmpart コマンドオプションはもうサポートがありません。**partx(8)** と **delpart(8)** のコマンドが今回、この機能を提供します。

3.5. テープデバイス

Red Hat Enterprise Linux 6.4 の時点で対応テープドライブ数は **128** から **512** に増えています。

第4章 ネットワーク設定とサービス

4.1. インターフェースと設定

NetworkManager

ネットワークインターフェースを設定する際、Red Hat Enterprise Linux 6 では NetworkManager が使用されますが、ネットワーク設定ツールの **system-config-network**、**system-config-network-tui**、**system-config-network-cmd** などにも収納されています。

Infiniband

Infiniband サポート (特に **openib** 開始スクリプトと **openib.conf** ファイル) は Red Hat Enterprise Linux 5 では **openib** パッケージによって提供されていました。このパッケージ名はその機能をより正確に反映するために Red Hat Enterprise Linux 6 で変更されています。Infiniband 機能は **rdma** パッケージ内で配布されるようになります。そのサービスは **rdma** という名前になり、設定ファイルは **/etc/rdma/rdma.conf** に配置されます。

biosdevname

biosdevname はカーネルデバイス名を引数として受け取ると BIOS で割り当てられたデバイス名を返します。デバイスの BIOS 名が直接あるいは確実にカーネル名にマッピングされないようなシステムで役に立ちます。**biosdevname** の使用は BIOS でデバイス名の割り当てが提供されるシステムやアーキテクチャーに限ってください。**biosdevname** ベースの命名をアクティブにするにはインストール時に **biosdevname=1** を渡します。

4.2. サービスの初期化

4.2.1. xinetd

Xinetd を使用してオンデマンドでネットワークサービスを開始します。xinetd 内の変更は **open** ファイルディスクリプタの許容限度に関連してきます。

- リスニングのメカニズムが **select()** から **poll()** に変更されています。この変更により、xinetd で使用される **open** ファイルディスクリプタの限度は変更可能になります。
- ファイルディスクリプタの限度がサービス単位でも変更できるようになります。**rlimit_files** ディレクティブを使ってサービスの設定ファイル内で行います。値は正の整数か UNLIMITED になります。

4.2.2. ランレベル

Red Hat Enterprise Linux 6 では、カスタムのランレベル 7、8、および 9 には対応しなくなるため使用できなくなります。

4.2.3. Upstart

Red Hat Enterprise Linux 6 では、**sysvinit** パッケージの **init** はイベントベースの **init** システムである **Upstart** に入れ替わっています。このシステムはブート時のサービスやタスクの開始、シャットダウン時の停止、システムの稼働中にはサービスやタスクの監視を行います。Upstart 自体に関する詳細は **init(8) man** ページを参照してください。

プロセスは **Upstart** でジョブとして認識されており、**/etc/init** ディレクトリー内のファイル群によって定義されます。Upstart については **man** ページ内詳しくが記載があります。コマンドの概要については **init(8)**、ジョブの構文については **init(5)** に説明があります。

Red Hat Enterprise Linux 6 の Upstart では、以下のような動作変更があります。

- **/etc/inittab** ファイルは非推奨となるため、**initdefault** 行でデフォルトのランレベルを設定する場合にしか使用されなくなります。その他の設定は **/etc/init** ディレクトリー内の **upstart** のジョブで行われます。
- アクティブな **tty** コンソール数は **/etc/sysconfig/init** 内の **ACTIVE_CONSOLES** 変数でセットされるようになります。この変数は **/etc/init/start-ttys.conf** ジョブによって読み込まれます。デフォルト値は **ACTIVE_CONSOLES=/dev/tty[1-6]** となり、**tty1** から **tty6** 上で **getty** を起動します。
- シリアルコンソールがシステムの主要コンソールになる場合、シリアル **getty** も自動的に設定されます。旧リリースでは、**/etc/inittab** を編集する **kudzu** によって行われていました。Red Hat Enterprise Linux 6 では、主要シリアルコンソールの設定は **/etc/init/serial.conf** で行われます。
- デフォルト以外のシリアルコンソールで実行している **getty** を設定するには、**/etc/inittab** を編集するのではなく、Upstart ジョブを記述する必要があります。例えば、**ttyS1** 上の **getty** を設定したい場合は、以下のようなジョブファイル (**/etc/init/serial-ttyS1.conf**) になるでしょう。

```
# This service maintains a getty on /dev/ttyS1.

start on stopped rc RUNLEVEL=[2345]
stop on starting runlevel [016]

respawn
exec /sbin/agetty /dev/ttyS1 115200 vt100-nav
```

旧リリースと同じく、この **getty** で **root** ログインを許可したい場合は **/etc/securetty** に **ttyS1** が必要になります。

Upstart への移動のため、**/etc/shutdown.allow** を使用してマシンのシャットダウンを行うことができるユーザーを定義する方法はサポートされなくなります。

4.3. IP テーブル/ファイアウォール

IPTable には **SECMARK** ターゲットモジュールが含まれています。SELinux などのセキュリティサブシステムで使用されるパケットに関連付けられるセキュリティマーク値の設定に使用されます。mangle 表内でのみ有効となります。使用例については以下を参照してください。

```
iptables -t mangle -A INPUT -p tcp --dport 80 -j SECMARK --selctx \
system_u:object_r:httpd_packet_t:s0
```

4.4. APACHE HTTP SERVER

Red Hat Enterprise Linux 6 への移行の際、注意すべき Apache HTTP Server の違いについて以下に示します。

- `mod_file_cache`、`mod_mem_cache`、`mod_imagemap` のモジュールには対応しなくなります。
- `Charset=UTF-8` オプションがデフォルトの `IndexOptions` ディレクティブに追加されています。UTF-8 以外の文字セットによるディレクトリ表示を必要とする場合には (`mod_autoindex` で生成されるものなど) このオプションを変更する必要があります。
- `distcache` で配信されるセッションキャッシュは `mod_ssl` では対応しなくなります。
- プロセス ID (pid) のデフォルトの位置が `/var/run` から `/var/run/httpd` に移動しています。
- `mod_python` パッケージはアップストリームでの開発が中止されたため同梱されなくなります。Red Hat Enterprise Linux 6 では代替として `mod_wsgi` を同梱させ、WSGI インターフェースによる Python スクリプト記述に対応します。

4.5. SAMBA

Samba 3.6 が Red Hat Enterprise Linux 6.4 に収録されています。主な変更点は以下の通りです。

- SMB2 サポート
- ID マッピングのリライト
- SMB Traffic Analyzer

詳細については Samba のリリースノートをご覧ください (<http://www.samba.org/samba/history/samba-3.6.0.html>)。

4.6. PHP

PHP の変更点を以下に示します。

- PHP はバージョン 5.3 にアップグレードされています。互換性に関する問題によりスクリプトの更新を必要とする場合があります。詳細については以下の URL を参照してください。
 - <http://php.net/manual/migration52.php>
 - <http://php.net/manual/migration53.php>
- デフォルト設定に次の変更が行われています (`/etc/php.ini`)。
 - `error_reporting` が `E_ALL & ~E_DEPRECATED` に設定されるようになります (以前は `previously E_ALL`)。
 - `short_open_tag` が `Off` に設定されるようになります (以前は `On`)。
 - `variables_order` が `GPCS` に設定されるようになります (以前は `EGPCS`)。
 - `enable_dl` が `Off` に設定されるようになります (以前は `On`)。
- `mime_magic`、`dbase`、`ncurses` の拡張は配信されなくなります。

4.7. BIND

BIND 設定には主要な変更点がいくつかあります。

デフォルトの ACL 設定

Red Hat Enterprise Linux 5 では、デフォルトの ACL 設定では全ホストに対して問い合わせを許可し、また再帰的な問い合わせを提供していました。Red Hat Enterprise Linux 6 のデフォルトでは、信頼すべきデータに対する問い合わせは全ホストが行えますが、再帰的な問い合わせについてはローカルネットワークのホストしか行うことはできません。

新しい `allow-query-cache` オプション

`allow-recursion` オプションは非推奨となり、このオプションが使用されるようになっていきます。信頼できないデータ全て (再帰的な検索や `root` ネームサーバーのヒントなど) を含むサーバーのキャッシュへのアクセスを制御します。

Chroot 環境の管理

`bind-chroot-admin` スクリプトは `chroot` 以外の環境から `chroot` 環境への `symlink` の作成に使用されていましたが、廃止されなくなります。代わりに設定は `chroot` 以外の環境で直接管理ができるようになります。このため、ファイルがまだ `chroot` に存在しない場合、`named` の起動中に `init` スクリプトにより必要なファイルが `chroot` 環境に自動的にマウントされます。

`/var/named` ディレクトリーのパーミッション

`/var/named` ディレクトリーには書き込みができなくなります。書き込みを必要とするゾーンファイル (動的 DNS ゾーン、DDNS など) はすべて書き込み可能となる新しいディレクトリー `/var/named/dynamic` に配置するようにしてください。

`dnssec [yes|no]` オプションの削除

グローバルなオプション `dnssec [yes|no]` は、新たに `dnssec-enable` と `dnssec-validation` の 2 種類のオプションに分割されています。`dnssec-enable` オプションは DNSSEC サポートを有効にします。`dnssec-validation` オプションは DNSSEC 検証を有効にします。`dnssec-enable` を再帰的サーバーで「no」に設定すると、DNSSEC 検証を行う別のサーバーではフォワーダーとしては使用できないということになります。いずれのオプションもデフォルトでは「yes」にセットされています。

`controls` ステートメント

`rndc` 管理ユーティリティを使用している場合は、`/etc/named.conf` 内で `controls` ステートメントを指定する必要がなくなります。`named` サービスは自動的にループバックデバイスを介して制御の接続を許可し、`named` と `rndc` の両方がインストール時に生成された同じシークレットキー (`/etc/rndc.key` 内にある) を使用します。

デフォルトのインストールでは、BIND は DNSSEC 検証が有効の状態です。ISC DLV レジスターを使用します。つまり、キーを ISC DLV レジスター内に持つ署名付きドメイン (`gov.`、`se.`、`cz.` など) はすべて再帰的サーバー上で暗号化検証が行われます。キャッシュポイズニングの試行により検証が失敗した場合、エンドユーザーにはこの偽のなりすましデータは与えられません。Red Hat Enterprise Linux 6 では DNSSEC の導入に完全対応しています。DNSSEC はエンドユーザーにとってインターネットをより安全に利用できるようにするための重要なステップとなります。前述のように DNSSEC 検証は `/etc/named.conf` 内の `dnssec-validation` オプションを使って制御します。

4.8. SNMP

Red Hat Enterprise Linux 6.4 およびそれ以前では、`Net-SNMP` の設定ファイルはシステム上の全ユーザーに対して読み取り可能で収納されていました。設定ファイルにはパスワードなど機密情報が格納される可能性があるため、Red Hat Enterprise Linux 6.5 から設定ファイルは `root` ユーザーにしか読み取りできないようになります。

この変更は **SNMP** 設定ファイルの `/etc/snmp/snmpd.conf` にアクセスを試行するユーザーのスクリプトに影響します。

4.9. NTP

NTP (Network Time Protocol) はコンピュータシステムのクロックをネットワーク経由で同期するのに使用されます。Red Hat Enterprise Linux 6 では、デフォルトの `/etc/ntp.conf` 設定ファイル内の次の行がコメントアウトされています。

```
#server 127.127.1.0 # local clock
#fudge 127.127.1.0 stratum 10
```

この設定は、**NTP** サーバーか参照クロックに明確に同期される場合にのみ、**ntpd** がネットワーククライアントに時刻情報を配信するということになります。同期されない場合でも **ntpd** が時間情報を提供するようにするには、上記 2 行のコメントを外して有効にする必要があります。

また、**ntpd** を `-x` オプション (`/etc/sysconfig/ntpd` ファイル内の **OPTIONS** で設定) で開始した場合または `/etc/ntp/step-tickers` 内にサーバーが指定されている場合、サービスは起動前に **ntpdate** コマンドを実行しなくなります。**ntpd** サービスとは別に単独で有効にできる **ntpdate** サービスが使用できるようになります。この **ntpdate** サービスはデフォルトでは無効になっています。使用は他のサービスが起動前に正確な時刻を必要とする場合に限ることを推奨します。これ以外の場合、後で時刻調節が **ntpd** によって実施された時に正常に機能しなくなります。

デフォルトの **NetworkManager** 構成でこのサービスを実行すると問題に遭遇する場合があります。Red Hat Enterprise Linux 導入ガイドに説明してあるように、**NETWORKWAIT=1** を `/etc/sysconfig/network` に追加すると修正できることがあります。

Red Hat Enterprise Linux 6.5 から **syslog** の **ntpd** で使用される形式が変更されています。この変更はユーザーがログメッセージの解析を試行する際に影響します。また、ユーザーは `ntp.conf` 内の **logconfig** オプションを使うとログ記録されるメッセージのタイプを設定できるようになります。

4.10. PTP

PTPv2 (IEEE 1588 Precision Time Protocol Version) はイーサネットのネットワーク内で正確なクロック同期を行うために使用されます。ハードウェアサポートと併用するとサブマイクロ秒領域でのクロックの正確性を実現することができます。Red Hat Enterprise Linux 6.5 以降、Red Hat では **PTP** に完全対応するため **Broadcom (bnx2x, tg3)**、**Intel (e1000e, igb, ixgbe)**、**Solarflare (sfc)** などのネットワークドライバと互換性を持つようになります。

4.11. KERBEROS

Red Hat Enterprise Linux 6 では、**Kerberos** のクライアントとサーバー (**KDC** を含む) は **des-cbc-crc**、**des-cbc-md4**、**des-cbc-md5**、**des-cbc-raw**、**des3-cbc-raw**、**des-hmac-sha1**、**arcfour-hmac-exp** などの暗号用キーを使用しないようデフォルト設定されます。デフォルトではクライアントはこのようなタイプのキーを持つサービスに対しては認証を行うことができません。

ほとんどのサービスのキータブに新しいキーセット (より強力な暗号で使用するキーも含む) を追加することができるため、ダウンタイムを発生することがありません。また、サービスのキーを与えるチケットも同様に **kadmin** の **cpw -keepold** コマンドを使用してより強力な暗号で使用するキーを含んだセットに更新することができます。

弱い暗号の使用を継続する必要があるシステムは、一時的な迂回策として `/etc/krb5.conf` ファイル内の `libdefaults` セクションに `allow_weak_crypto` オプションを必要とします。この変数は、デフォルトでは `false` にセットしてあるため、このオプションを有効にしないと認証は失敗します。

```
[libdefaults]
allow_weak_crypto = yes
```

また、使用可能な共有ライブラリとしても、アプリケーション内でのサポートされている認証メカニズムとしても、`Kerberos IV` のサポートは削除されています。ロックアウトポリシー用に新たに追加されたサポートはデータベースダンプ形式に変更が必要となります。旧式の `KDC` が使用できる形式でデータベースをダンプする必要があるマスターの `KDC` には、`-r13` オプションを付けた `kdb5_util` の `dump` コマンドを実行させる必要があります。

4.12. MAIL

4.12.1. Sendmail

Red Hat Enterprise Linux 5 の一部のリリースでは、`sendmail` MTA (Mail Transport Agent) がデフォルトで外部ホストからのネットワーク接続を受け取っていました。Red Hat Enterprise Linux 6 では、`sendmail` はデフォルトではローカルのシステム (`localhost`) からの接続しか受け付けません。リモートホストに対してサーバーとして動作できる機能を `sendmail` に与える場合は、以下の手順のいずれかを行います。

- `/etc/mail/sendmail.mc` を編集して、`DAEMON_OPTIONS` 行を変更しネットワークデバイス上でもリッスンするようにします
- `/etc/mail/sendmail.mc` の `DAEMON_OPTIONS` 行に `#` 印を入れてコメントアウトします。

上記いずれかの変更を反映させるには、`sendmail-cf` パッケージをインストールしてから `/etc/mail/sendmail.cf` を再生成します。次のコマンドを実行して行います。

```
su -c 'yum install sendmail-cf'
su -c 'make -C /etc/mail'
```

4.12.2. Exim

`Exim` は Red Hat Enterprise Linux 6 から削除されています。Postfix がデフォルトで推奨される MTA になります。

4.12.3. Dovecot

4.12.3.1. Dovecot の設定

Dovecot 2.x の構成が変更されています。マスター設定ファイル `/etc/dovecot.conf` が `/etc/dovecot/dovecot.conf` に移動されているため、Dovecot 構成の他の部分も `/etc/dovecot/conf.d/*.conf` に移動されています。構成の大部分は同じであり、この新しいバージョンとの互換性があります。以下のコマンドを使用するとご使用の構成をテストして、この新バージョンで名前変更されているオプション、削除されているオプション、変更されているオプションを一覧表示させることができます。

```
doveconf [-n] -c /old/dovecot.conf
```

4.13. MYSQL®

4.13.1. DBD ドライバー

MySQL DBD ドライバーは二重ライセンスとなり、ライセンスに関連する問題は解決されました。結果、`apr-util-mysql` パッケージが Red Hat Enterprise Linux 6 のソフトウェアレポジトリに収納されるようになります。

4.14. POSTGRESQL

4.14.1. データベースをアップグレードする

PostgreSQL 8.4 (`postgresql84-*` パッケージ) を使用していた既存の Red Hat Enterprise Linux 5 からアップグレードを行っている場合は、Red Hat Enterprise 6 PostgreSQL パッケージが当座の代替として動作するようになります。

ただし、PostgreSQL 8.1 (`postgresql-*` パッケージ) またはこれより旧式のバージョンを使用していた Red Hat Enterprise Linux 5 インストールからのアップグレードで、維持しておく必要があるデータベースコンテンツが既存している場合、データ形式が変更されているためダンプに従い記載されている手順を再ロードする必要があります (<http://www.postgresql.org/docs/8.4/interactive/install-upgrading.html>)。必ずダンプの手順を先に行ってから、Red Hat Enterprise Linux 6 のアップグレードを行ってください。

4.14.2. その他の変更

PostgreSQL 8.1 から 8.4 への移行に関連して考えられるアプリケーションの互換問題については、<http://wiki.postgresql.org/wiki/WhatsNew84> を参照してください。

4.15. SQUID

Squid はバージョン 3.1 に更新されています。この更新によりネイティブの IPv6 サポートを提供するようになり、設定ファイル `/etc/squid/squid.conf` が大幅に短縮されています。ただし、設定オプションにも大幅な変更があるため、一部の旧バージョンとは完全な下方互換がありません。変更についての詳細は Squid 3.1 リリースノートを参照してください (<http://www.squid-cache.org/Versions/v3/3.1/RELEASENOTES.html>)。

4.16. BLUETOOTH

4.16.1. オンデマンドの Bluetooth サービス

Bluetooth デバイスに対応するため、Red Hat Enterprise Linux の旧バージョンではデフォルトで Bluetooth バックグラウンドサービスが起動していました。本リリースでは、Bluetooth サービスは必要に応じてオンデマンドで起動し、デバイスの使用が停止されてから 30 秒後に自動で停止します。これにより全体的な初期起動時間とリソース消費が低減されます。

4.17. CRON

4.17.1. Vixie cron と Cronie

Red Hat Enterprise Linux 6 には `vixie-cron` の代替として `crone` パッケージが含まれています。これらのパッケージの主な違いは、定期的なジョブ (毎日、毎週、毎月) の実行方法にあります。`Cronie` は `/etc/anacrontab` ファイルを使用します。このファイルのデフォルト状態を以下に示します。

```
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45

# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

# period in days    delay in minutes    job-identifier      command

1    5    cron.daily nice run-parts /etc/cron.daily
7    25   cron.weekly nice run-parts /etc/cron.weekly
@monthly 45   cron.monthly nice run-parts /etc/cron.monthly
```

これらの定期的なジョブはランダム遅延を含め 03:00 から 22:00 の時間帯で毎日一回実行されます。例えば、`cron.daily` は 5 分の強制遅延とそれに加えて 0 分から 45 分のランダム遅延があります。4 時から 5 時の間で遅延なくジョブを実行することもできます。

```
RANDOM_DELAY=0 # or do not use this option at all

START_HOURS_RANGE=4-5

# period in days    delay in minutes    job-identifier      command

1    0    cron.daily nice run-parts /etc/cron.daily
7    0    cron.weekly nice run-parts /etc/cron.weekly
@monthly 0    cron.monthly nice run-parts /etc/cron.monthly
```

`crone` の機能を以下に示します。

- `/etc/anacrontab` 内でのジョブ開始のランダムな遅延が可能です。
- 定期的なジョブの時間幅を `/etc/anacrontab` で定義することができます。
- 各 `cron` テーブルに `CRON_TZ` 変数を付けそれ自体に定義したタイムゾーンを持たせることができます。
- デフォルトでは `cron` デーモンは `inotify` でテーブル内の変更をチェックします。

`crone` と `crone-anacron` の詳細については Red Hat Enterprise Linux 導入ガイドを参照してください。

4.18. ログ記録

`dateext` オプションが `/etc/logrotate.conf` 内でデフォルトで有効になります。このオプションは、日付 (YYYYMMDD 形式) を表す拡張を追加することによりログファイルの古いバージョンをアーカイブします。以前はファイルに番号が追記されていました。

第5章 コマンドラインツール

本セクションでは、Red Hat Enterprise Linux 6 の各コマンドラインツールの動作変更について説明しています。

5.1. GREP

`grep` コマンドの動作では、大文字と小文字の文字列検索について変更が加えられています。[a-z] 形式の範囲検索の使用は、`LC_COLLATE` 変数に依存します。

`LC_COLLATE=C` とセットすると旧来の動作を保存することができるため、この方法で範囲検索を実行した場合に正しい結果を得ることができます。しかし、Red Hat Enterprise Linux 6 では、範囲検索には `[:lower:]`、`[:upper:]` 形式の使用をお勧めします。

この変更は出力に多大な影響を及ぼします。そのため、正しい結果を引き続き得ることができるようスクリプトとプロセスの再確認を行ってください。

5.2. SED

`-i` オプションを付けて `sed` コマンドを使用すると読み取り専用ファイルの内容、および他の保護ファイルの削除ができます。任意のファイルのパーミッションはそのファイルに対して行うことができる動作を定義し、任意のディレクトリのパーミッションはそのディレクトリ配下のファイル群の一覧に対して行うことができる動作を定義します。このことから `sed` では読み取り専用のディレクトリ内にある書き込み可能なファイル上では `-i` を使用できません。また、このようなファイル上で `-i` オプションを使用するとシンボリックリンクやハードリンクが破損することになります。

5.3. PCRE

`pcre` パッケージは 7.8 に更新されています。これには以下のような動作変更が含まれています。

- UTF-8 チェックは RFC 2279 ではなく RFC 3629 を参照するようになります。これにより、受け取る文字列での制約が厳しくなります。例えば、UTF-8 文字の序数値は `0x0010FFFF` に制限されるようになります。

```
$ echo -ne "\x00\x11\xff\xff" | recode UCS-4-BE..UTF8 | pcregrep --utf-8 '.'  
pcregrep: pcre_exec() error -10 while matching this line:
```

RFC の詳細については <http://tools.ietf.org/html/rfc3629#section-12> を参照してください。

- PCRE の旧バージョンでコンパイルした保存済みのパターンは再コンパイルが必要です。これは事前にコンパイルした PCRE 式を外部メモリ (1 ファイルなど) に直列化して後それを読み込むようなアプリケーションに影響を及ぼします。通常、大規模のスパムフィルタなどパフォーマンスを向上させる目的で行われます。

5.4. シェル

シェルのバイナリファイル群の場所が変更されています。例えば、`bash` バイナリと `ksh` のバイナリは `/usr/bin` からなくなります。いずれのバイナリも `/bin` に配置されるようになります。バイナリの新しい場所をポイントするようスクリプトを更新する必要があります。

5.5. NAUTILUS

nautilus-open-terminal パッケージでは、右クリックによる **Open Terminal** オプションで現在のディレクトリ内に新規のターミナルウィンドウを開きます。以前は、**Desktop** からこのオプションを選択すると、新規のターミナルウィンドウの位置はユーザーのホームディレクトリに設定されていました。Red Hat Enterprise Linux 6 のデフォルト動作では **Desktop** ディレクトリが開きます (**~/Desktop/**)。以前の動作を有効にする場合は、次のコマンドを使って **desktop_opens_home_dir** GConf boolean を **true** にセットします。

```
gconftool-2 -s /apps/nautilus-open-terminal/desktop_opens_dir --type=bool  
true
```

第6章 システム設定

6.1. ACPI

6.1.1. CPU のホットプラグ

Red Hat Enterprise Linux 6.5 では ACPI CPU のホットプラグはテクノロジープレビューとして利用することができます。この機能はプラットフォーム固有の機能となります。使い方については本ガイドの範疇外となります。

重要

テクノロジープレビューの機能により次回の製品機能に早くアクセスできるため、お客様は開発段階で機能性をテストし、フィードバックを提供することができます。ただし、テクノロジープレビューの機能は Red Hat Enterprise Linux Subscription Level Agreement では完全にサポートされていず機能的に完全ではない可能性があるため、実稼働での使用を目的としていません。テクノロジープレビュー機能の開発段階では、追加のコンポーネントがテスト向けに公開されることがあります。テクノロジープレビュー機能は開発中であるため、Red Hat では機能の安定性については保証できません。このため、テクノロジープレビュー機能を使用している場合、同機能の後続リリースへのスムーズなアップグレードが行えない場合があります。

Red Hat は今後リリースするテクノロジープレビュー機能については完全にサポートする予定ですが、この機能がエンタープライズ向けに実用化するための基準を満たしていないことが明らかになる場合もあります。そのような場合には、テクノロジープレビュー機能がサポート対象の機能としてリリースされる保証はありません。テクノロジープレビュー機能の中には、特定のハードウェアアーキテクチャーでのみ利用可能なものもあります。

この機能を使用するには `CONFIG_ACPI_HOTPLUG_CPU` 設定オプションを有効にする必要があります。

また、プラットフォームでオプションの `ACPI_OST` メソッドを実装する場合、以下の設定オプションについても有効にする必要があります。いずれのプラットフォームの場合でもこの設定を有効にすることで有害となることはありません。

- `CONFIG_ACPI_HOTPLUG_CPU`
- `CONFIG_ACPI_HOTPLUG_MEMORY` または `CONFIG_ACPI_HOTPLUG_MEMORY_MODULE`
- `CONFIG_ACPI_CONTAINER` または `CONFIG_ACPI_CONTAINER_MODULE`

6.2. ログ記録

`dateext` オプションが `/etc/logrotate.conf` 内でデフォルトで有効になります。このオプションは、日付 (YYYYMMDD 形式) を表す拡張を追加することによりログファイルの古いバージョンをアーカイブします。以前はファイルに番号が追記されていました。

第7章 デスクトップ

Red Hat Enterprise Linux 6 では GUI コンソールが `tty7` から `tty1` に移動しています。

7.1. GDM の設定

GDM 設定がいくつか GConf 内で管理されるようになります。

GDM のデフォルト Greeter はシンプルな Greeter と呼ばれ、GConf で設定します。デフォルト値は `gdm-simple-greeter.schemas` ファイル内の GConf に保存されています。この値を編集する場合は、`gconftool2` か `gconf-editor` を使用します。Greeter には以下のオプションがあります。

- `/apps/gdm/simple-greeter/banner_message_enable`

`false` (Boolean)

バナーのテキストメッセージが表示されるかどうかを制御します。

- `/apps/gdm/simple-greeter/banner_message_text`

`NULL` (string)

greeter ウィンドウで表示するバナーのテキストメッセージを指定します。

- `/apps/gdm/simple-greeter/logo_icon_name`

`computer` (string)

greeter ログに使用するテーマアイコン名を設定します。

- `/apps/gdm/simple-greeter/disable_restart_buttons`

`false` (Boolean)

ログインウィンドウに再起動ボタンを表示するかどうかを制御します。

- `/apps/gdm/simple-greeter/wm_use_compiz`

`false` (Booleans)

`metacity` ではなく `compiz` をウィンドウマネージャとして使用するかどうかを制御します。

GConf を使用するとプラグインも無効にすることができます。例えば、サウンドのプラグインを無効にしたい場合、以下のキーを設定解除します：`/apps/gdm/simple-greeter/settings-manager-plugins/sound/active`

第8章 セキュリティと認証

本章では、SELinux、SSSD、LDAP、Checksum、PAMなどのセキュリティおよび認証に関する動作変更について説明しています。

8.1. SELINUX

sshd デーモンは制限のあるサービスとなります。

8.2. SSSD

プロバイダーと呼ばれるリモートの識別メカニズムおよび認証メカニズムへのアクセスを提供するのが SSSD (System Security Services Daemon) です。SSSD を使用しこうしたプロバイダーを SSSD のバックエンドとして設定すると実際の (ローカルおよびネットワーク上の) 識別および認証ソースを取り出すことができるようになります。また、あらゆる種類の識別データプロバイダーをプラグインすることが可能です。ユーザー情報を含んでいるデータベースとなるドメインはプロバイダーの識別情報ソースとして機能します。複数の識別情報プロバイダーに対応するため、2つ以上の識別サーバーを別々のユーザーネームスペースとして動作させることが可能です。収集された情報は標準の PAM や NSS インターフェースを介しフロントエンドとなるアプリケーションで利用することができます。

SSSD は複数のサービスから成るセットとして稼働し、それを使用するアプリケーションには依存しません。このため、アプリケーション自体にリモートのドメインへの接続を持たせる必要がなく、また使用しているドメインを認識する必要もありません。識別情報およびグループメンバーシップ情報をローカルにキャッシングできる堅牢な機能により識別情報元 (LDAP、NIS、IPA、DB、Samba など) に関係無く運用が可能になり、パフォーマンスが向上されるためオフラインで稼働していてオンライン認証が利用できない場合であっても認証を行うことができます。また、SSSD では同じタイプの複数のプロバイダー (複数の LDAP プロバイダー) を使用できるため、ドメイン修飾型の識別情報要求を別のプロバイダーで解決することができます。詳細については「Red Hat Enterprise Linux 6 導入ガイド」をご覧ください。

8.3. LDAP

8.3.1. OpenLDAP

OpenLDAP サービスに必要な設定が Red Hat Enterprise Linux 6 で変更になっています。旧バージョンでは、slapd は `/etc/openldap/slapd.conf` で設定していました。Red Hat Enterprise Linux 6 の slapd 設定は特殊な LDAP ディレクトリー (`/etc/openldap/slapd.d/`) に事前定義済みのスキーマ、DIT (ディレクトリー情報ツリー) などと合わせて格納されるようになります。この設定スキーマの詳細については openldap.org をご覧ください。以下のセクションでは、新しいディレクトリーで古い設定ファイルが機能するよう変換する方法について例をあげて説明していきます。

8.3.2. slapd 設定を変換する

この例では、古い slapd 設定から変換するファイルが `/etc/openldap/slapd.conf` にあり、OpenLDAP 設定の新しいディレクトリーは `/etc/openldap/slapd.d/` にあると仮定しています。

- 新しいディレクトリーの `/etc/openldap/slapd.d/` の内容を削除します。

```
# rm -rf /etc/openldap/slapd.d/*
```

- `slaptest` を実行して設定ファイルの有効性をチェックし、新しい設定ディレクトリーを指定します。

```
slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d
```

- 新しいディレクトリーでパーミッションを設定します。

```
chown -R ldap:ldap /etc/openldap/slapd.d
```

```
chmod -R 000 /etc/openldap/slapd.d
```

```
chmod -R u+rwX /etc/openldap/slapd.d
```

- サービスが新しい設定ディレクトリー内で機能していることを確認してから、古い設定ファイルを削除します。

```
rm -rf /etc/openldap/slapd.conf
```

8.4. チェックサム

Red Hat Enterprise Linux ではデータ検証や認証を行う際以前に比べより多岐に渡って SHA-256 ダイジェストアルゴリズムが使用され、暗号的に強度が弱い SHA-1 および MD5 アルゴリズムからアップグレードしています。

8.5. PAM (プラグ可能な認証モジュール)

PAM サービス用の共通設定は `/etc/pam.d/system-auth-ac` ファイルに配置されています。

認証モジュールは追加の PAM 設定ファイル、`/etc/pam.d/password-auth-ac`、`/etc/pam.d/smartcard-auth-ac`、`/etc/pam.d/fingerprint-auth-ac` などにも書き込まれるようになります。

Red Hat Enterprise Linux 6 では、`sshd` の PAM モジュールおよび `ftpd` などの他のリモートサービスには `/etc/pam.d/system-auth` ファイルではなく `/etc/pam.d/password-auth` ファイルが収納されるようになります。

8.6. システムのユーザー

Red Hat Enterprise Linux 6 では、静的に割り当てられる UID/GID 番号のしきい値 (`/usr/share/doc/setup-*/uidgid` ファイル内の `setup` パッケージで定義される) が 100 (Red Hat Enterprise Linux 3、4、5) から 200 に増えています。このため、動的または静的に割り当てられた 100 から 200 までの UID/GID を持っているシステムの場合はその影響を受け、一部のアプリケーションのインストールおよび実行に障害が発生する可能性があります。

Red Hat Enterprise Linux 6 では動的な UID/GID の割り当て範囲は 499 以下になります。静的なシステムユーザーを作成する際に `setup` パッケージによる強制的な予約番号を持たせない場合は、300 またはそれ以上の UID/GID 番号を使用することをお勧めします。

8.7. NIST SCAP 1.2 証明書

Red Hat Enterprise Linux 6.6 以降は、OpenSCAP (`openscap`) は NIST (National Institute of Standards and Technology) の SCAP 1.2 (Security Content Automation Protocol) で認証されるようになります。SCAPでは、自動的にパッチの存在を検証する、システムのセキュリティ構成の設定を確認する、不正

アクセスの兆候がないかシステムを検査する、などエンタープライズ向けシステムの安全性を維持できるよう標準化された手段を提供します。

また、Red Hat Enterprise Linux 6.6 には新しいパッケージ `scap-security-guide` も収納され、OpenSCAP の最適な使い方が説明されています。

8.8. RSA および DSA キーの生成に関する変更点

Red Hat Enterprise Linux 6 の通常の操作では RSA および DSA キー生成に関するサイズの規制はありません。Red Hat Enterprise Linux 6 を FIPS モードで稼働する場合には制限が適用されます。

Red Hat Enterprise Linux 6.6 以降、FIPS モードでのキー生成動作は `OPENSSL_ENFORCE_MODULUS_BITS` の環境変数で指定されることになります。

FIPS モードが使用され `OPENSSL_ENFORCE_MODULUS_BITS` 環境変数が設定されている場合は生成できる RSA および DSA キーは 2048 ビットまたは 3072 ビットのみになります。

`OPENSSL_ENFORCE_MODULUS_BITS` 環境変数を設定していない場合はキー生成の動作は Red Hat Enterprise Linux 6 の旧リリースと変わらず、1024 ビット以上の RSA キーおよび 1024 ビット、2048 ビット、3072 ビットの DSA キーを作成することができます。

第9章 システムのモニタリングとカーネル

9.1. DRACUT

`mkinitrd` は `dracut` ツールに置き換えられました。また、`/etc/modprobe.conf` ファイルはデフォルトではカーネルモジュールの管理には使用されなくなります。ただし、手作業で作成した場合にはまだ使用することができます。`dracut` ツールの使用例については以下を参照してください。

```
# mv /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r)-old.img
# dracut --force /boot/initramfs-$(uname -r).img $(uname -r)
```

`dracut` の設定ファイルは `/etc/dracut.conf.d` 設定ディレクトリーをご覧ください。

9.2. IPVS (IP VIRTUAL SERVER)

Red Hat Enterprise Linux 6.5 では新しい同期メッセージプロトコル形式を導入しています。途中でタイムアウトしてしまっているバックアップサーバーのノードですっと永続的に接続が行われてしまうことが原因でフェールオーバーの際にそのバックアップサーバーの状態が不整合になることがありました。新しい同期メッセージプロトコル形式では、停止できないような重要なビジネスサービスにこのようなことが発生しないよう防ぎます。

この新しい形式は Red Hat Enterprise Linux 6.4 およびそれ以前のバージョン、または `kernel-2.6.32-406.el6` 以前のカーネルバージョンとの互換性はありません。マスターノードのアップグレードを行う前に、バックアップノードのアップグレードを行ってください。

バックアップノードより先にマスターノードのアップグレードが必要な場合など、同期メッセージに旧形式の使用を継続する場合は `/proc/sys/net/ipv4/vs/sync_version` の値を `0` に設定します。

9.3. ジョイスティックのサポート

ジョイスティックデバイスのサポートはデフォルトでは有効になりません。Red Hat Enterprise Linux 6 カーネルではジョイスティックモジュールを提供しなくなります。

9.4. ABRT (自動バグ報告ツール)

Red Hat Enterprise Linux 6.6 以降、ABRT (バグ報告ツール) はバージョン 2 に更新されています。この更新により制約が一部取り除かれますが、設定および動作に変更点があります。

クラッシュなどに伴う問題データがデータベースに格納されなくなります。この情報は問題データディレクトリーにしか格納されなくなります。これにより `/etc/abrt/abrt.conf` 設定ファイルが簡略化され設定ディレクティブの一部が廃止予定になる、または別の場所で指定されることとなります。

- `OpenPGPCheck`、`BlackList`、`ProcessUnpackaged`、`BlackListedPaths` のディレクティブ `/etc/abrt/abrt.conf` ファイルではなく、`/etc/abrt/abrt-action-save-package-data.conf` ファイルで指定するようになります。
- `Database` ディレクティブは不要になるため対応しなくなります。
- `ActionsandReporters` ディレクティブは `post-create` イベントに置き換えられています。ABRT 2 でのイベントに関する詳細は「[ABRT イベント](#)」を参照してください。
- `abrt.conf` ファイルの `[AnalyzerActionsAndReporters]` セクションは非推奨になります。以前、このセクションで設定されていたディレクティブ (`Kerneloops`、`CcCpp`、`Python`)

は **analyze_*** と **report_*** のイベントに置き換えられています。イベントの詳細については「[ABRT イベント](#)」を参照してください。

- C/C++ のフック機能は **ReadonlyLocalDebugInfoDirs** ディレクティブを除き **abrt-ccpp** サービスに置き換えられています。 **ReadonlyLocalDebugInfoDirs** ディレクティブはまだポートされていません。
- Python のフック機能は **abrt-addon-python** パッケージに置き換えられています。
- **oops** のフック機能は **abrt-oops** サービスおよび関連する **abrt-dump-oops** と **abrt-action-kerneloops** のコマンドに置き換えられています。

ABRT では問題データを柔軟に自動化して報告できるよう以下のようなコマンドを用意しています。

reporter-bugzilla

指定問題データのディレクトリを読み取り、同じ ABRT ハッシュを持つバグがないか **Bugzilla** を検索します。同じハッシュを持つバグが見つかった場合、そのバグにコメントが残されます。同じ ABRT ハッシュを持つバグがなかった場合は新しいバグが作成されます。デフォルトのパッケージの他、**libreport-plugin-bugzilla** パッケージも追加が必要とします。

reporter-kerneloops

カーネルの **oops** を適切なサイトに報告します。デフォルトのパッケージの他、**libreport-plugin-kerneloops** パッケージも追加が必要とします。

reporter-mailx

メールで問題データのディレクトリーのコンテンツを送信します。デフォルトのパッケージの他、**libreport-plugin-mailx** プラグインも追加が必要とします。

reporter-print

問題データを標準の出力または指定したファイルに表示します。デフォルトのパッケージの他、**libreport-plugin-logger** パッケージも追加が必要とします。

reporter-rhtsupport

問題データを RHT サポートに報告します。デフォルトのパッケージの他、**libreport-plugin-rhtsupport** プラグインも追加が必要とします。

reporter-upload

問題データのディレクトリーの **tarball** を指定した URL にアップロードします。デフォルトのパッケージの他、**libreport-plugin-reportuploader** パッケージも追加が必要とします。

9.4.1. ABRT イベント

ABRT 2 では ABRT のワークフローに設定可能なイベントが追加されます。問題データが記録されるとイベントが発生します。イベントによって問題データに対し行う動作が特定されます。データの分析方法を変更したりデータのアップロード先を指定したりすることもできます。また、特定の特性を持つ問題データが記録された場合にのみイベントが発生するよう設定することもできます。

イベントの設定ファイルは **/etc/libreport/events.d** ディレクトリーに格納されます。次のようなものが含まれます。

イベント名

発生させるイベントの名前です。これは1番目の引数となる **EVENT** パラメーターです。たとえば、次のイベント設定ファイルには **report_Bugzilla** という名前のイベントが含まれています。

```
EVENT=report_Bugzilla analyzer=Python
reporter-bugzilla -c /etc/libreport/plugins/Bugzilla.conf
```

条件

指定した条件が問題データに一致する場合、このイベントに指定した動作が問題データに実行されます。この例では、**Python** の値を含んでいる **analyzer** ファイルが問題データのディレクトリーに格納されている場合にのみイベントが発生します。

```
EVENT=report_Bugzilla analyzer=Python
reporter-bugzilla -c /etc/libreport/plugins/Bugzilla.conf
```

動作

このイベントが発生する問題データに対して行われる動作です。この場合、**reporter-bugzilla** コマンドが実行されます。

```
EVENT=report_Bugzilla analyzer=Python
reporter-bugzilla -c /etc/libreport/plugins/Bugzilla.conf
```

詳しくは **man** ページを参照してください。

```
$ man report_event.conf
```

第10章 パッケージとドライバーに関する変更点

収納されているパッケージとシステムドライバー群は Red Hat Enterprise Linux のリリースの度に定期的な変更を受けます。変更が行われる理由として、「新しい機能を提供するためオペレーティングシステムにパッケージやドライバーが追加または更新される」、「パッケージやドライバーが旧式で無効なハードウェア対応となるため削除される」、「パッケージやドライバーのアップストリームプロジェクトが維持されなくなる」、「ハードウェアベンダーによるハードウェア固有のパッケージやドライバーの対応が行われなくなるため削除される」などがあげられます。

本章では、Red Hat Enterprise Linux 6 で更新されたもの、新規となるもの、また非推奨となり廃止予定になったもの、実際に廃止されたもの(削除)について記載しています。

10.1. システム設定ツールの変更点

10.1.1. system-config-bind

system-config-bind ツールは非推奨となっているため削除され、代替ツールはありません。Red Hat Enterprise Linux 6 では、ネームサーバーの設定を編集する場合は **named.conf** ファイルから手作業で行うことをお勧めします。BIND に関する総合的なガイドは **bind** パッケージの一部として **/usr/share/doc/bind-x.y.z** 内にインストールされます。また、設定例については **/usr/share/doc/bind-x.y.z/sample** ディレクトリーをご覧ください。ただし、旧バージョンの system-config-bind ツールは標準の BIND 設定を生成するため、ご使用の環境によっては古い設定ファイルを正しい場所に移動して十分なテストを実行すれば Red Hat Enterprise Linux 6 にある BIND のバージョンに移行することも可能です。

10.1.2. system-config-boot

system-config-boot ツールでは GRUB ブートローダーのグラフィカルな設定が可能でした。Red Hat Enterprise Linux 6 では、グラフィカルな設定は非推奨となっているため削除され、代替となるものはありません。多くのユーザーにとってはデフォルトの GRUB 設定で十分なはずですが、手作業による変更が必要な場合には **/boot/grub** ディレクトリー内の **grub.conf** ファイルに起動に関する設定がありますので、ここで変更することができます。Red Hat Enterprise Linux 6 では GRUB レガシーとも呼ばれる GRUB のバージョン 1 を使用しています。GRUB の設定に関する詳細については GRUB のホームページ <http://www.gnu.org/software/grub/> をご覧ください。

10.1.3. system-config-cluster

system-config-cluster ツールは非推奨となっているため削除され、代替のツールはありません。ricci と luci (Conga プロジェクトのパッケージ) の使用をお勧めします。

10.1.4. system-config-display

system-config-display ツールは、対応デスクトップとなる GNOME および KDE の両方で XRandr 設定ツールに入れ替わっています。ディスプレイ管理は以下のメニューオプションのいずれかで動的に行われるようになるため、デフォルトの X サーバーインストール内には明示的な設定ファイル (**xorg.conf**) はありません。

GNOME: システム → 設定 → ディスプレイ (または **system-config-display** コマンド)

KDE: システムの設定 → コンピュータ管理 → ディスプレイ

ディスプレイ設定にコマンドラインユーティリティ (**xrandr**) を使用することもできます。詳細については **xrandr --help** コマンドまたは **man xrandr** コマンドで **man** ページを参照してください。

10.1.5. system-config-httpd

`system-config-httpd` ツールは非推奨となっているため削除され、代替となるツールはありません。Web サーバーの設定は手作業で行ってください。設定は `/etc/httpd` ディレクトリで行うことができます。主要設定ファイルは `/etc/httpd/conf/httpd.conf` にあります。このファイル内には、ほとんどのサーバー設定に関して詳細な説明が記載されています。ただし、必要であれば Apache Web サーバーに関して詳細に記載されているドキュメントが `httpd-manual` パッケージ内に収納されています。

10.1.6. system-config-lvm

`system-config-lvm` ツールは非推奨となっています。論理ボリュームの管理については `gnome-disk-util` または `lvm` のツールで行うことができます。

10.1.7. system-config-netboot

`system-config-netboot` ツールは非推奨となっているため削除され、代替のツールはありません。Red Hat Network Satellite の使用をお勧めします。

10.1.8. system-config-nfs

`system-config-nfs` ツールは非推奨となっているため削除され、代替のツールはありません。NFS サーバーの設定は手作業で行う必要があります。

10.1.9. system-config-rootpassword

`system-config-rootpassword` ツールは、ユーザーの管理と設定を行うことができるパワフルなツール `system-config-users` ツールに入れ替わっています。root パスワードは `system-config-users` ツール内で設定できます。個人設定ダイアログ内の「システムのユーザーとグループを隠す」オプションのチェックを外します。root ユーザーが主要一覧に表示されるようになり、他のユーザーと同じようにパスワードを変更することができるようになります。

10.1.10. system-config-samba

`system-config-samba` ツールは非推奨となるため削除され、代替のツールはありません。SMB サーバーの設定は手作業で行う必要があります。

10.1.11. system-config-securitylevel

`system-config-securitylevel` ツールは削除されています。ファイアーウォールの設定には `system-config-firewall` ツールの使用をお勧めします。

10.1.12. system-config-soundcard

`system-config-soundcard` ツールは削除されています。サウンドカードの検出と設定は自動的に行われます。

10.1.13. system-config-switchmail

`system-config-switchmail` ツールは非推奨となっているため削除され、代替のツールはありません。Red Hat Enterprise Linux 6 では Postfix がデフォルトの MTA (メール転送エージェント) で推奨となります。別の MTA を使用している場合は、その MTA 固有の設定ファイルと技術に応じた設定を手作業で行う必要があります。

10.1.14. 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、実際のアップグレードプロセス開始前に、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、パッケージの削除や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、以下のアイテムを提供します。

- 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は、<https://access.redhat.com/site/node/637583/> を参照してください。

10.1.15. Red Hat Upgrade Tool

新しい **Red Hat Upgrade Tool** は **Preupgrade Assistant** の後に使用され、アップグレードプロセスの以下の 3 つの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** パッケージ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。
- **Red Hat Upgrade Tool** がアップグレード後にクリーンアップを実行し、アップグレードされたオペレーティングシステムにシステムを再起動します。

ネットワークベースおよびディスクベースの両方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指示については、<https://access.redhat.com/site/node/637583/> を参照してください。

10.2. BASH (BOURNE-AGAIN SHELL)

Red Hat Enterprise Linux 6 には **Bash** のバージョン 4.1 がデフォルトシェルとして収納されています。本セクションではバージョン 4.1 によりもたらされる旧バージョンとの互換性に関する問題点について説明しています。

- **Bash-4.0** およびそれ以降のバージョンは、プロセス置換の構成をブレース展開を使って変更せずに渡すことができるようになるため、内容の展開はすべて別々に指定し、各プロセス置換を別々に入力しなければならなくなります。
- **Bash-4.0** およびそれ以降のバージョンでは **Posix** が指定するのと同様に、**SIGCHLD** が **wait** の組み込みに割り込みを許可するため、全ての子を待機するよう「**wait**」を使用する場合、子を終了したら **SIGCHLD** トラップは常に呼び出されることがなくなります。

- **Bash-4.0** およびそれ以降のバージョンは、クローズ用の区切り文字 `$()` コマンド置換を検索する場合に **Posix** のルールに従うようになり、旧バージョンのような動作はしなくなりますが、より多くの構文および解析のエラーを先に捕らえてからコマンド置換を評価するためのサブシェルを生成します。
- プログラム可能な完了コードはコマンドラインを単語に分解する際にシェルのメタキャラクターのセットではなく読み込み行と同じ区切り文字のセットを使用します。これにより、プログラム可能な完了と読み込み行の一貫性が増します。
- **read** のビルトインが時間切れになると、入力の **read** を指定変数に割り当てようとするため、十分な入力がない場合には変数が空の文字列にセットされる原因にもなります。旧バージョンは文字の **read** を破棄していました。
- **Bash-4.0** およびそれ以降のバージョンでは、パイプラインのコマンドのひとつがコマンド一覧を実行している間に **SIGINT** によって終了させられた場合、シェルは割り込みを受けたかのような動作をします。
- **Bash-4.0** およびそれ以降のバージョンでは、**set -e** オプションの処理法が変更されるため、パイプラインが失敗すると (失敗したパイプライン内の最後のコマンドが単純なコマンドでない場合も) シェルは終了します。これは **Posix** が指定するものとは異なります。この部分の基準を更新する作業が進展中です。 **Bash-4.0** の動作はリリースの時点での合意を得ようとしている動作です。
- **Bash-4.0** およびそれ以降のバージョンでは、`."` がシステムの **PATH** に存在しない場合でも、**.(source)** ビルトインがファイル名の引数を現在のディレクトリで検索してしまう原因となっていた **Posix** モードのバグが修正されています。 **Posix** では、このような場合シェルによる **PWD** 変数内の検索は行われるべきではないと述べられています。
- **Bash-4.1** は、`[[` コマンドに演算子を使用して文字列を比較する場合は現在のロケールを使用します。 **compatNN shopt** オプションの1つをセットすることで以前の動作に戻すことができます。

10.2.1. 正規表現

既に記載されている点に加えて、正規表現に対するパターン引数に引用符を付けて条件演算子 `=~` に一致させると、正規表現の一致が動作しなくなる場合があります。これは全てのアーキテクチャで起こります。 **3.2** 以前のバージョンとなる **bash** では、`[[` コマンドの `=~` 演算子に対する正規表現の引数に引用符を付けることによる効果は規定されていませんでした。実用的な効果としては、パターン引数に二重引用符を付けると、バックスラッシュ `/` で特殊パターン文字に引用符を付けなければなりませんでしたが。これにより、二重引用符が付けられた単語展開で行われるバックスラッシュの処理を妨害し、`==` シェルパターン一致演算子が引用符が付けられた文字を処理する方法と矛盾していました。

bash バージョン **3.2** では、`=~` 演算子に対して一重の引用符が付けられた文字列引数および二重の引用符が付けられた文字列引数内の文字には内部的に引用符が付けられるようシェルが変更されました。これにより、`(「\。」、「\['」、「\」、「\('、\）」、「*」、「\+」、「\?」、「\{」、「\|」、「\^」、「\$_」)` を処理する正規表現に重要となる文字の特殊な意味を抑制し、そのまま強制的に一致が行われるようにします。`==` パターン一致演算子とそのパターン引数内の引用符が付けられた部分を処理する方法との矛盾がなくなります。

引用符付き文字列引数の処理が変更されてから問題がいくつか出現してきています。主要な問題として、パターン引数内の空白、**bash 3.1** と **bash 3.2** 間で引用符付き文字列の処理が異なるなどの問題があげられます。いずれの問題もシェル変数を使ってパターンを維持することで解決できます。`[[` コマンドの全オペランドでシェル変数を展開する場合には単語の分割は行われないため、変数を割り当てる時に好きなようにパターンに引用符を付けてから、空白を含む可能性がある単独の文字列にその値を展開することができるようになります。最初の問題は、バックスラッシュか他の引用メカニズムを使用してパターン内で空白をエスケープすることで解決します。

Bash 4.0 は `shopt` ビルトインに対するいくつかのオプションで制御される互換性レベルの概念を取り入れています。 `compat31` オプションを有効にすると、 `bash` は `=~` 演算子の右側の引用に関して 3.1 の動作を取り戻します。

10.3. その他のパッケージ変更点

10.3.1. 新しいパッケージ

ここでは Red Hat Enterprise Linux 6 の新しいパッケージについて説明しています。

java-1.8.0-openjdk



重要

OpenJDK 8 はテクノロジープレビューとして Red Hat Enterprise Linux 6.6 で提供されます。テクノロジープレビュー機能に関するサポートについては <https://access.redhat.com/site/support/offerings/techpreview/> を参照してください。

Red Hat Enterprise Linux 6.6 では OpenJDK 8 はテクノロジープレビューとして提供しています。OpenJDK 8 は JSR 337 で定義されている Java SE 8 プラットフォーム仕様のオープンソース参照実装です。この新しいバージョンの仕様によりラムダ式と Java プログラミング言語に対するデフォルトのメソッドが追加されています。また、日付と時刻の新しい API も収納され、タイプにアノテーションが行えるよう Java プログラミング言語が拡張されています。

本リリースでは Javadoc 解析がより厳密になっています。つまり、旧バージョンの OpenJDK 用に作成した javadocs の一部はコンパイルできない場合があります。変更に応じた変更を javadocs で修正する前にコンパイルが必要な場合は javadocs 生成を一時的に無効にすると厳密な要件を回避することができます。

また、このリリースから非推奨になるメソッドがいくつかあります。詳細については仕様を確認してください。

libnl3

libnl3 パッケージでは netlink インターフェース用ライブラリーの集合となる libnl のバージョン 3 を提供しています。libnl3 は libnl と同じような機能を提供していますが、libnl の後方互換性を持たせる代替ではありません。このため、Red Hat Enterprise Linux 6.6 からは libnl3 はオリジナルの libnl パッケージと平行インストールすることができます。

redhat-support-tool

Red Hat サポートツール (redhat-support-tool) を使用するとサブスクライブサービスへのコンソールベースでのアクセスが容易になり、Red Hat のお客様向けコンテンツやサービスへのアクセスポイントを広げました。さらに、ヘルプデスクサービスとサブスクリプションサービスを統合、自動化することもできるようになります。本パッケージの機能には以下のようなものがあります。

- Red Hat アクセスナレッジベースの記載やソリューションをコンソールを使用して表示できます (man ページの形式)。
- コンソールを使用してカスタマーサポートケースを表示したり、また作成、変更、コメント入力なども行なうことができます。
- コンソールを使用し、カスタマーサポートケースや <ftp://dropbox.redhat.com/> に添付ファイルを直接アップロードすることができます。

- プロキシに完全対応します (FTP および HTTP プロキシ)。
- コンソールを使用して、添付ファイルを一覧表示させカスタマーサポートケースにダウンロードすることができます。
- クエリー用語、ログメッセージ、その他パラメータなどで Red Hat アクセスナレッジベースを検索し、選択可能な一覧で検索結果を表示させることができます。
- 分析を行なうため Red Hat アクセス自動問題判別エンジンにログファイル、テキストファイル、その他のソースなどを容易にアップロードすることができます。
- これ以外にもさまざまなサポート関連のコマンドが用意されています。

ツールの使い方についての詳細は Red Hat カスタマーポータルをご覧ください (<https://access.redhat.com/site/articles/445443>)。

redhat-support-lib-python

Red Hat のお客様は本ライブラリーを使用してヘルプデスクのソリューションや IT インフラストラクチャーを Red Hat カスタマーポータル提供のサービスと統合することができます。さらに、自動問題判別、診断のため同梱製品への統合にも使用されます。

本パッケージで提供される python ライブラリーは Red Hat カスタマーポータルとのやりとりを簡略化する抽象層になります。必要な許可証明書を与え API のインスタンスを作成し、Red Hat カスタマーポータルとの通信にその API オブジェクトを使用するだけです。

API に対応できる通信の一例として、ログファイルの自動診断サービス、ナレッジベースの検索、サポートケースの作成、サポートケースへのファイルの添付、サポートケースの状態を表示、エンタイトルメントの表示などがあります。

rsyslog7

新しい rsyslog7 パッケージでは rsyslog バージョン 7 を提供しています。旧パッケージの rsyslog では継続してバージョン 5 を提供しています。

Red Hat Enterprise Linux 6.6 から Red Hat は rsyslog7 パッケージのメンテナンスに焦点を置くようになるため、rsyslog パッケージでは安全性に関する重大な問題しか修正されなくなります。これによりバージョン 5 を必要とするレガシーなプロジェクトにも対応しながら、パフォーマンスが改善され体系的なログ機能が導入されたバージョン 7 を提供することができるようになります。

10.3.2. 更新パッケージ

Red Hat Enterprise Linux 6 で更新されているパッケージ、注目すべき変更点について説明しています。

表10.1 更新パッケージ

更新パッケージ	説明
OProfile	OProfile は 0.9.5 に更新されています。この新しいバージョンには、Intel Atom および i7 のプロセッサ、AMD Family 11h のプロセッサ、AMD Family 10h の IBS (Instruction Based Sampling) 機能が含まれています。

更新パッケージ	説明
quota、edquota、setquota	ユーザー名またはユーザー ID を引数として受け取るようになります。引数が数字で表される場合はユーザー ID とみなされます。これ以外は ID に自動的に変換されます。ユーザー名が数字のみで構成されているような場合には問題となる可能性があるため注意してください。 quota パッケージが更新されています。 quota 、 edquota 、 setquota などのユーティリティでユーザー名を強制的に ID に変換していた -x 引数は削除されています。この機能は --always-resolve オプションで提供されるようになります。
module-init-tools	/etc/modprobe.conf はデフォルトでは存在しません。手作業で作成すればまだ使用可能です。
openscap	Red Hat Enterprise Linux 6.6 から openscap パッケージは SCAP 1.2 に準じてリベースされています。コンポーネント仕様に対して複数の変更が行われています。詳細は http://scap.nist.gov をご覧ください。

10.3.3. 廃止パッケージ

Red Hat Enterprise Linux 6 で廃止 (削除) されているパッケージとその代替パッケージについて説明します。

表10.2 廃止パッケージ

廃止パッケージ	代替パッケージ
aspell	hunspell 、 aspell はビルド依存関係としてのみ提供されます。スペルチェックを使用するアプリケーションは hunspell を使用しなければなりません。
beecrypt	NSS/OpenSSL
crash-spu-commands	代替のパッケージなし。Cell 固有のパッケージは含まれなくなります。
dhcpv6/dhcpv6-client	dhcp/dhclient のバイナリには IPv6 対応機能が組み込まれるようになります。
elfspe2	代替のパッケージなし。Cell 固有のパッケージは含まれなくなります。
exim	Postfix
gnbd	代用として iSCSI のご使用を推奨します。

廃止パッケージ	代替パッケージ
gnome-vfs	gvfs
ipsec-tools	Openswan
kmod-gnbd	代用として iSCSI のご使用を推奨します。
lam	openmpi
libatomic	gcc-libraries、Red Hat Enterprise Linux 6.5 およびそれ以降
libspe2	代替のパッケージなし。Cell 固有のパッケージは含まれなくなります。
libspe2-devel	代替のパッケージなし。Cell 固有のパッケージは含まれなくなります。
linuxwacom	xorg-x11-drv-wacom
mod_python	mod_wsgi、Python スクリプトの記述には代替として WSGI インターフェースを使用する mod_wsgi を使用することができます。
mkinitrd	dracut
nss_ldap	nss-pam-ldapd、pam_ldap
openmotif-2.2	openmotif-2.3
spu-tools	代替のパッケージなし。Cell 固有のパッケージは含まれなくなります。
switchdesk	セッションの管理は、サポートされている 2 種類のセッションマネージャ GDM と KDM で行われます。
sysklogd	rsyslog (Red Hat Enterprise Linux 6.0 以降)、rsyslog7 (Red Hat Enterprise Linux 6.6 以降)
SysVinit	upstart
vixie-cron	cronie

10.3.4. 非推奨パッケージ

- qt3

- GFS1
- gcj、パフォーマンス確保のため Red Hat Enterprise Linux 6 には収納されますが、今後のリリースでは収納されない可能性があります。
- cryptoloop
- mingw32-qpidd-cpp
- python-qmfd
- python-qpidd
- qpidd-cpp
- qpidd-qmfd
- qpidd-tests
- qpidd-tools
- ruby-qpidd
- saslwrapper

10.4. ドライバー関連の変更点

本セクションでは、Red Hat Enterprise Linux 6 でのドライバー関連の変更点を説明します。ドライバーはすべてデフォルトで `initramfs` にロードされる点に注意してください。

10.4.1. 廃止ドライバー

- aic7xxx_old
- atp870u
- cpqarray
- DAC960
- dc395x
- gdth
- hfs
- hfsplus
- megaraid
- net/tokenring/
- paride
- qla1280
- sound/core/oss

- sound/drivers/opl3/*
- sound/pci/nm256

10.4.2. 非推奨ドライバー

- aacraid
- aic7xxx
- i2o
- ips
- megaraid_mbox
- mptlan
- mptfc
- sym53c8xx

10.4.3. 廃止カーネルコンポーネント

- NBD - Network Block Device は Red Hat Enterprise Linux 6 で iSCSI に置き換えられました。
- HFS - Apple ファイルシステムのサポートは Red Hat Enterprise Linux 6 で廃止されました。
- Tux - Web Server アクセラレーターは Red Hat Enterprise Linux 6 で廃止されました。
- PAE 非搭載の x86 カーネル - Red Hat Enterprise Linux の旧バージョンでは i686 アーキテクチャー用に複数のカーネルを収納していました (PAE 搭載のカーネルと非搭載のカーネル)。PAE 非搭載のハードウェアが大量に販売されてから長い年月を経ています。したがって、Red Hat Enterprise Linux 6 では PAE を搭載したカーネルがひとつのみになります。
- 先行の I/O スケジューラは非推奨となり Red Hat Enterprise Linux 6 からなくなります。CFQ (Completely Fair Queueing) I/O スケジューラが代替となります。このスケジューラは 2006 年から Linux カーネル内のデフォルトの I/O スケジューラでした。先行の I/O スケジューラをご使用の方は CFQ で作業負荷のテストを実施してパフォーマンス関連の問題が見られる場合にはバグを提出されることをお勧めします。CFQ があらゆるテスト用作業負荷下で先行の I/O スケジューラと同等のパフォーマンスを行えることが目標となりますが、すべてのテストが期待通りでこの目標から外れるようなパフォーマンスはないとは保証できません。

10.5. ライブラリ関連の変更点

32-bit のライブラリはデフォルトでは Red Hat Enterprise Linux 6 にインストールされません。`/etc/yum.conf` で `multilib_policy=all` をセットするとこの動作を変更することができます。これにより、`multilib` ポリシーがシステム全域のポリシーとして有効になります。

付録A 改訂履歴

改訂 6.2-9.2 翻訳ドラフト	Fri Feb 6 2015	
改訂 6.2-9.1 翻訳ファイルを XML ソースバージョン 6.2-9 と同期	Fri Feb 6 2015	
改訂 6.2-9 RHEL 6.6 GA 向けにビルド	Wed Oct 08 2014	Laura Bailey
改訂 6.2-8 一点を説明している別々のアイテムを結合	Thu Aug 18 2014	Laura Bailey
改訂 6.2-7 sysklogd 代替パッケージの名前を訂正、BZ1088684	Mon Aug 11 2014	Laura Bailey
改訂 6.2-6 RHEL 6.6 Beta にそってドキュメントのリリース準備	Fri Aug 08 2014	Laura Bailey
改訂 6.2-5 libnl3 と libnl の平行インストールに関するセクション内の言い回しを更新、BZ1092776	Tue Jul 29 2014	Laura Bailey
改訂 6.2-4 Preupgrade Assistant BZ1088147 と Red Hat Upgrade Tool BZ1087196 に関するセクションを追加	Tue Jul 29 2014	Laura Bailey
改訂 6.2-3 rsyslog7 パッケージに関する説明、rsyslog のメンテナンスに対する変更点についての記載を追加、BZ1088684 新しい libnl3 パッケージに関する記載を追加、BZ1092776	Mon Jul 28 2014	Laura Bailey
改訂 6.2-1 RSA および DSA キーの生成に関する変更点の記載を追加、BZ1088154 (tmraz による確認済み)	Wed Jun 11 2014	Laura Bailey
改訂 6.2-0 Preupgrade Assistant に関するセクションを追加、BZ1088147 Red Hat Upgrade Tool に関するセクションを追加、BZ1087196	Wed Jun 04 2014	Laura Bailey
改訂 6.1-92 システムのモニタリングとカーネルの章内の ABRT セクションを訂正、BZ710098	Wed May 27 2014	Laura Bailey
改訂 6.1-91 システムのモニタリングとカーネルの章内の ABRT セクションに対する記載を追加、BZ710098	Mon May 19 2014	Laura Bailey
改訂 6.1-89 Squid バージョン 3.1 への更新に関する間違った記載を削除、BZ1020649 openscap の SCAP 1.2 証明書を追加、BZ1092775 OpenJDK8 テクノロジープレビューに関する記載を追加および訂正、BZ1088664 システムのモニタリングとカーネルの章に ABRT セクションを追加、BZ710098 非推奨となった qpid パッケージについて記載、BZ977643	Fri May 16 2014	Laura Bailey
改訂 6.1-83 RHEL 6.4 と RHEL 6.5 間での LVS 同期の相互運用性に関する問題について記載、BZ#1008708 SME のフィードバックを反映し更新	Wed Sep 18 2013	Laura Bailey
改訂 6.1-81 RHEL 6.5 に関する SME の最終フィードバックを反映	Thu Sep 05 2013	Laura Bailey

改訂 6.1-80 SME のフィードバックを反映 QE のフィードバックを反映	Tue Aug 13 2013	Laura Bailey
改訂 6.1-77 SME のフィードバックを反映 新しい <code>redhat-support-lib-python</code> パッケージについて記載 (BZ#986093) 新しい <code>redhat-support-tool</code> パッケージについて記載 (BZ#986106) NTP のログフォーマットに関する変更点について記載 (BZ#986774)	Fri Aug 09 2013	Laura Bailey
改訂 6.1-74 ソースコードの整理 システム設定の章を追加 SNMP 設定ファイルへのアクセスに関する変更について記載 (BZ#977584) ACPI CPU ホットプラグのテクノロジープレビューに関する詳細を追加 (BZ#970475) PTP のドライバーサポートに関する詳細を追加 (BZ#970474) <code>libatomic</code> の代替パッケージとなる <code>gcc-libraries</code> について記載 (BZ#970441)	Thu Aug 01 2013	Laura Bailey
改訂 6.1-69 <code>cryptoloop</code> が非推奨となる点を記載	Wed Nov 21 2012	Scott Radvan
改訂 6.1-68 Samba 3.6 およびリリースノートへのリンクについて記載	Sun Oct 14 2012	Scott Radvan
改訂 6.1-67 テープデバイスの制限について修正、 BZ#847907	Mon Sep 10 2012	Scott Radvan
改訂 6.1-66 報告のあった誤字の修正、 BZ#853204	Mon Sep 3 2012	Scott Radvan
改訂 6.1-65 <code>dracut.conf.d</code> 設定ディレクトリに関する記載を追加	Mon Sep 3 2012	Scott Radvan
改訂 6.1-64 誤字修正	Mon Aug 27 2012	Scott Radvan
改訂 6.1-63 ガイド全体に ID タグを追加	Mon Aug 27 2012	Scott Radvan
改訂 6.1-62 対応テープドライブに関する変更を追加	Mon Aug 27 2012	Scott Radvan
改訂 6.1-61 6.3 GA リリースに向け公開	Mon Jun 18 2012	Scott Radvan
改訂 6.1-59 <code>fusecompress</code> のセクションを除外、 BZ#791258	Fri Feb 17 2012	Scott Radvan
改訂 6.1-58 誤字修正、 BZ#664683	Mon Jan 16 2012	Scott Radvan
改訂 6.1-57 デフォルトのカーネルではジョイスティックのサポートが提供されない旨の記載、 BZ#664683	Mon Jan 16 2012	Scott Radvan
改訂 6.1-55 6.2 リリース向け査読	Mon Nov 28 2011	Scott Radvan

改訂 6.1-39

6.1 リリース用に点検

Wed May 18 2011

Scott Radvan