



Red Hat Enterprise Linux 9

RHEL インストーラーのブートオプション

起動オプションを指定してインストールプログラムの動作をカスタマイズする

Red Hat Enterprise Linux 9 RHEL インストーラーのブートオプション

起動オプションを指定してインストールプログラムの動作をカスタマイズする

法律上の通知

Copyright © 2024 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

RHEL インストールシステムには、管理者向けのさまざまなブートオプションが含まれています。これらのオプションを使用すると、インストールプログラムをカスタマイズする特定の機能を有効または無効にして、インストールプログラムのデフォルトの動作を変更できます。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	3
第1章 起動オプション	4
1.1. 起動オプションの入力	4
1.2. 起動オプションの編集	4
1.3. インストールソースの起動オプション	5
1.4. ネットワーク起動オプション	10
1.5. コンソール起動オプション	14
1.6. 起動オプションのデバッグ	16
1.7. ストレージ起動オプション	18
1.8. キックスタート起動オプション	18
1.9. 高度なインストール起動オプション	20
1.10. 削除済みの起動オプション	20

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

Jira からのフィードバック送信 (アカウントが必要)

1. [Jira](#) の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. ダイアログの下部にある **Create** をクリックします。

第1章 起動オプション

本セクションは、インストールプログラムのデフォルトの挙動を変更するのに使用できる起動オプションの一部を説明します。すべての起動オプションは、アップストリームの [Boot Options](#) を参照してください。

1.1. 起動オプションの入力

起動オプションには、等号 (=) が付いているものと、付けていないものがあります。ブートオプションはブートコマンドラインに追加され、スペースで区切って複数のオプションを追加できます。インストールプログラムに固有の起動オプションは、常に **inst** から始まります。

等号 (=) 記号を使用するオプション

起動オプションに、= 記号を使用する値を指定する必要があります。たとえば、**inst.vncpassword=** オプションには値 (この場合はパスワード) を指定する必要があります。この例の正しい構文は **inst.vncpassword=password** です。

等号 (=) 記号を使用しないオプション

この起動オプションでは、値またはパラメーターを使用できません。たとえば、**rd.live.check** オプションでは、インストール開始前にインストールメディアの検証が強制されます。インストールプログラムは、このブートオプションが存在すると検証を実行します。ブートオプションが存在しないと、検証はスキップされます。

1.2. 起動オプションの編集

このセクションでは、ブートメニューからブートオプションを編集するさまざまな方法について説明します。インストールメディアを起動すると、起動メニューが開きます。

1.2.1. BIOS で boot: プロンプトの編集

boot: プロンプトを使用すると、最初のオプションは、読み込むインストールプログラムのイメージファイルを常に指定する必要があります。ほとんどの場合、このイメージはキーワードを使用して指定できます。要件に応じて、追加オプションを指定できます。

前提条件

- 起動可能なインストールメディア (USB、CD、または DVD) を作成している。
- メディアからインストールを起動し、起動メニュー画面が開いている。

手順

1. ブートメニューが開いたら、キーボードの **Esc** キーを押します。
2. **boot:** プロンプトにアクセスできるようになります。
3. キーボードの **Tab** キーを押して、ヘルプコマンドを表示します。
4. キーボードの **Enter** キーを押して、オプションでインストールを開始します。**boot:** プロンプトから起動メニュー画面に戻るには、システムを再起動して、インストールメディアから再度起動します。



注記

boot: プロンプトでは、**dracut** カーネルオプションも使用できます。利用可能なオプションの一覧は、**dracut.cmdline(7)** の man ページを参照してください。

1.2.2. > プロンプトを使用して事前定義されたブートオプションの編集

BIOS ベースの AMD64 および Intel64 システムでは、> プロンプトを使用して、事前定義されたブートオプションを編集できます。オプションの完全なセットを表示するには、ブートメニューから **Test this media and install RHEL 9** を選択します。

前提条件

- 起動可能なインストールメディア (USB、CD、または DVD) を作成している。
- メディアからインストールを起動し、起動メニュー画面が開いている。

手順

1. ブートメニューでオプションを選択し、キーボードの **Tab** キーを押します。> プロンプトにアクセスし、利用可能なオプションを表示します。
2. > プロンプトに必要なオプションを追加します。
3. **Enter** を押してインストールを開始します。
4. **Esc** キーを押して編集をキャンセルし、ブートメニューに戻ります。

1.2.3. UEFI ベースのシステムの GRUB2 メニューの編集

GRUB2 メニューは、UEFI ベースの AMD64、Intel 64、および 64 ビット ARM システムで利用できません。

前提条件

- 起動可能なインストールメディア (USB、CD、または DVD) を作成している。
- メディアからインストールを起動し、起動メニュー画面が開いている。

手順

1. ブートメニューウィンドウから必要なオプションを選択し、**e** を押します。
2. UEFI システムでは、カーネルコマンドラインは **linuxefi** で始まります。カーソルを **linuxefi** カーネルコマンドラインの最後に移動します。
3. 必要に応じてパラメーターを編集します。たとえば、1つ以上のネットワークインターフェイスを設定するには、**linuxefi** カーネルコマンドラインの最後に **ip=** パラメーターを追加し、その後に必要な値を追加します。
4. 編集が終了したら、**Ctrl + X** を押して、指定したオプションを使用してインストールを開始します。

1.3. インストールソースの起動オプション

このセクションでは、さまざまなインストールソースのブートオプションについて説明します。

inst.repo=

inst.repo= 起動オプションはインストールソースを指定します。つまり、パッケージリポジトリと、そのリポジトリを記述する有効な **.treeinfo** ファイルを提供する場所にあたります。たとえば、**inst.repo=cdrom** になります。**inst.repo=** オプションの対象は、以下のいずれかのインストールメディアになります。

- インストール可能なツリー (インストールプログラムのイメージ、パッケージ群、リポジトリデータおよび有効な **.treeinfo** ファイルを含むディレクトリー設定)
- DVD (システムの DVD ドライブにある物理ディスク)
- Red Hat Enterprise Linux のフルインストール用 DVD の ISO イメージ (ディスク、またはシステムにアクセスできるネットワーク上の場所)

inst.repo= 起動オプションでは、さまざまなインストール方法を設定します。以下の表は、**inst.repo=** 起動オプションの詳細な構文を記載します。

表1.1 inst.repo= ブートオプションおよびインストールソースのタイプおよびフォーマット

ソースタイプ	起動オプションの形式	ソースの形式
CD/DVD ドライブ	inst.repo=cdrom:<device>	物理ディスクとしてのインストール DVD。 [a]
マウント可能なデバイス (HDD および USB スティック)	inst.repo=hd:<device>:/<path>	インストール DVD のイメージファイル
NFS サーバー	inst.repo=nfs: [options:]<server>:/<path>	インストール DVD のイメージファイル、またはインストールツリー (インストール DVD にあるディレクトリーおよびファイルの完全なコピー)。 [b]
HTTP サーバー	inst.repo=http://<host>/<path>	インストールツリー (インストール DVD 上にあるディレクトリーおよびファイルの完全なコピー)。
HTTPS サーバー	inst.repo=https://<host>/<path>	
FTP サーバー	inst.repo=ftp://<username>:<password>@<host>/<path>	
HMC	inst.repo=hmc	

ソースタイプ	起動オプションの形式	ソースの形式
	[a] device が省略された場合、インストールプログラムはインストール DVD を含むドライブを自動的に検索します。	
	[b] NFS サーバーのオプションでは、デフォルトで NFS プロトコルのバージョン 3 が使用されます。別のバージョンを使用するには、 nfsvers=X をオプションに追加し、X を、使用するバージョン番号に置き換えます。	

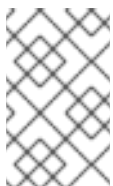
ディスクデバイス名は、次の形式で設定します。

- カーネルデバイス名 (例: **/dev/sda1** または **sdb2**)
- ファイルシステムのラベル (例: **LABEL=Flash** または **LABEL=RHEL8**)
- ファイルシステムの UUID (例: **UUID=8176c7bf-04ff-403a-a832-9557f94e61db**)

英数字以外は **\xNN** で表す必要があります。NN は文字の 16 進数表示になります。たとえば、**\x20** なら空白 (" ") になります。

inst.addrepo=

inst.addrepo= 起動オプションを使用して、別のインストールソースとして、メインリポジトリ (**inst.repo=**) とともに追加のリポジトリを追加します。起動時に、**inst.addrepo=** 起動オプションを複数回使用できます。以下の表では、**inst.addrepo=** 起動オプションの構文の詳細を記載します。



注記

REPO_NAME はリポジトリの名前であり、インストールプロセスでは必須です。これらのリポジトリは、インストールプロセス時にのみ使用され、インストールしたシステムにはインストールされません。

統一された ISO に関する詳細は、[Unified ISO](#) を参照してください。

表1.2 インストールソースおよびブートオプションの形式

インストールソース	起動オプションの形式	関連情報
URL にあるインストール可能なツリー	inst.addrepo=REPO_NAME, [http,https,ftp]://<host>/<path>	指定の URL にあるインストール可能なツリーを探します。
NFS パスにあるインストール可能なツリー	inst.addrepo=REPO_NAME,nfs://<server>/<path>	指定した NFS パスのインストール可能なツリーを探します。コロンは、ホストの後に必要です。インストールプログラムは、RFC 2224 に従って URL の解析を行うのではなく、 nfs:// ディレクトリ名の後のすべてを mount コマンドに渡します。

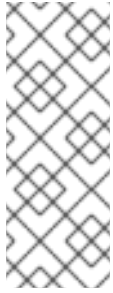
インストールソース	起動オプションの形式	関連情報
インストール環境でインストール可能なツリー	inst.addrepo=REPO_NAME,file://<path>	インストール環境の指定した場所にあるインストール可能なツリーを探します。このオプションを使用するには、インストールプログラムが利用可能なソフトウェアグループのロードを試行する前に、リポジリーがマウントされる必要があります。このオプションの利点は、起動可能な ISO に複数のリポジリーを利用でき、ISO からメインリポジリーと追加のリポジリーの両方をインストールできることです。追加のリポジリーへのパスは /run/install/source/REPO_ISO_PATH です。また、キックスタートファイルの %pre セクションにリポジリーディレクトリーをマウントできます。パスは、 inst.addrepo=REPO_NAME,file:///<path> など、/ で始まる必要があります。
ディスク	inst.addrepo=REPO_NAME,hd:<device>:<path>	指定した <device> パーティションをマウントして、<path> で指定した ISO からインストールします。<path> を指定しないと、インストールプログラムは <device> 上の有効なインストール ISO を探します。このインストール方法には、有効なインストール可能ツリーを持つ ISO が必要です。

inst.stage2=

inst.stage2= 起動オプションは、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。このオプションは、有効な **.treeinfo** ファイルが含まれるディレクトリーへのパスを想定し、**.treeinfo** ファイルからランタイムイメージの場所を読み取ります。**.treeinfo** ファイルが利用できないと、インストールプログラムは、**images/install.img** からイメージを読み込もうとします。**inst.stage2** オプションを指定しない場合、インストールプログラムは **inst.repo** オプションで指定された場所を使用しようとしています。

このオプションは、後でインストールプログラム内でインストールソースを手動で指定する場合に使用します。たとえば、インストールソースとしてコンテンツ配信ネットワーク (CDN) を選択する場合などに使用します。インストール DVD および Boot ISO には、それぞれの ISO からインストールプログラムを起動するための適切な **inst.stage2** オプションがすでに含まれています。

インストールソースを指定する場合は、代わりに **inst.repo=** オプションを使用します。



注記

デフォルトでは、インストールメディアで **inst.stage2=** 起動オプションが使用され、これは特定のラベル (たとえば **inst.stage2=hd:LABEL=RHEL-x-0-0-BaseOS-x86_64**) に設定されています。ランタイムイメージが含まれるファイルシステムのデフォルトラベルを修正する場合、またはカスタマイズされた手順を使用してインストールシステムを起動する場合は、**inst.stage2=** 起動オプションに正しい値が設定されていることを確認してください。

inst.noverifyssl

inst.noverifyssl 起動オプションを使用して、追加のキックスタートリポジトリを除き、すべての HTTPS 接続の SSL 証明書が検証されないようにします。ただし、**--noverifyssl** はリポジトリごとに設定できます。

たとえば、リモートのインストールソースが自己署名 SSL 証明書を使用している場合には、**inst.noverifyssl** 起動オプションは、SSL 証明書を検証せずにインストーラーがインストールを完了できるようにします。

inst.stage2= を使用してソースを指定する場合の例

```
inst.stage2=https://hostname/path_to_install_image/ inst.noverifyssl
```

inst.repo= を使用してソースを指定する場合の例

```
inst.repo=https://hostname/path_to_install_repository/ inst.noverifyssl
```

inst.stage2.all

inst.stage2.all 起動オプションを使用して、複数の HTTP、HTTPS、または FTP ソースを指定します。**inst.stage2=** 起動オプションは、**inst.stage2.all** オプションとともに複数回使用して、成功するまで、イメージを順番にフェッチできます。以下に例を示します。

```
inst.stage2.all
inst.stage2=http://hostname1/path_to_install_tree/
inst.stage2=http://hostname2/path_to_install_tree/
inst.stage2=http://hostname3/path_to_install_tree/
```

inst.dd=

インストール時にドライバーの更新を実行する場合は、**inst.dd=** 起動オプションを使用します。インストール時にドライバーを更新する方法の詳細は、[高度な RHEL 9 インストールの実行](#) を参照してください。

inst.repo=hmc

このオプションにより、外部ネットワーク設定の必要がなくなるため、インストールのオプションが増えます。Binary DVD から起動すると、インストーラープログラムにより、追加のカーネルパラメーターを入力するように求められます。DVD をインストールソースとして設定するには、**inst.repo=hmc** オプションをカーネルパラメーターに追加します。インストールプログラムは、サポート要素 (SE) およびハードウェア管理コンソール (HMC) のファイルアクセスを有効にし、DVD から stage2 のイメージをフェッチし、ソフトウェア選択のために DVD のパッケージへのアクセスを提供します。

inst.proxy=

HTTP、HTTPS、および FTP プロトコルからインストールを実行する場合には、**inst.proxy=** 起動オプションが使用されます。以下に例を示します。

■

[PROTOCOL://][USERNAME[:PASSWORD]@]HOST[:PORT]

inst.nosave=

inst.nosave= 起動オプションを指定して、インストールログや関連ファイルがインストール済みのシステムに保存されないように制御します (例: **input_ks**、**output_ks**、**all_ks**、**logs**、**all**)。複数の値をコンマで区切って組み合わせることができます。以下に例を示します。

```
inst.nosave=Input_ks,logs
```



注記

inst.nosave ブートオプションは、インストール済みのシステムから、キックスタートのログや入力/出力などの Kickstart %post スクリプトで削除できないファイルの除外に使用されます。

input_ks

キックスタートによる入力を保存する機能を無効にします。

output_ks

インストールプログラムで生成されたキックスタートによる出力を保存する機能を無効にします。

all_ks

キックスタートによる入出力を保存する機能を無効にします。

logs

すべてのインストールログを保存する機能を無効にします。

all

すべてのキックスタート結果とすべてのログを保存する機能を無効にします。

inst.multilib

inst.multilib 起動オプションを使用して、DNF の **multilib_policy** を、**best** ではなく **all** に設定します。

inst.memcheck

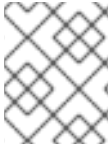
inst.memcheck 起動オプションは、インストールを完了するのにシステムに十分な RAM があることを確認するためのチェックを実行します。RAM が十分でない場合は、インストールプロセスが停止します。システムのチェックはおおよそのもので、インストールの際のメモリー使用率は、パッケージ選択やユーザーインターフェイス (グラフィカル、テキスト)、その他のパラメーターにより異なります。

inst.nomemcheck

inst.nomemcheck 起動オプションは、インストールを完了するのに十分な RAM があるかどうかの確認を実行しません。推奨よりも低いメモリー量でのインストールはサポートされていないため、インストールプロセスが失敗する場合があります。

1.4. ネットワーク起動オプション

シナリオでローカルイメージから起動するのではなく、ネットワーク経由でイメージから起動する必要がある場合は、次のオプションを使用してネットワーク起動をカスタマイズできます。



注記

dracut ツールを使用してネットワークを初期化します。**dracut** オプションの完全なリストについては、**dracut.cmdline(7)** の man ページを参照してください。

ip=

ip= 起動オプションは、1つ以上のネットワークインターフェイスを設定します。複数のインターフェイスを設定するには、次のいずれかの方法を使用します。

- インターフェイスごとに1回ずつ、**ip** オプションを複数回使用します。これを行うには、**rd.neednet=1** オプションを使用し、**bootdev** オプションを使用してプライマリーブートインターフェイスを指定します。
- **ip** オプションを1回使用してから、Kickstart を使用してさらにインターフェイスを設定します。このオプションでは、複数の形式が使用できます。以下の表は、最も一般的なオプションの情報が含まれます。

以下の表では、下記の点を前提としています。

- **ip** パラメーターはクライアントの IP アドレスを指定し、**IPv6** には角括弧が必要です (例: 192.0.2.1 または [2001:db8::99])。
- **gateway** パラメーターはデフォルトゲートウェイになります。**IPv6** には角括弧が必要です。
- **netmask** パラメーターは使用するネットマスクです。完全ネットマスク (255.255.255.0 など) または接頭辞 (64 など) を使用できます。
- **hostname** パラメーターはクライアントシステムのホスト名です。このパラメーターは任意です。

表1.3 ネットワークインターフェイスを設定するためのブートオプション形式

起動オプションの形式	設定方法
ip=method	全インターフェイスの自動設定
ip=interface:method	特定インターフェイスの自動設定
ip=ip::gateway:netmask:hostname:interface:none	静的設定 (例: IPv4 ip=192.0.2.1::192.0.2.254:255.255.255.0:server.example.com:enp1s0:none) IPv6: ip=[2001:db8::1]::[2001:db8::ffe]:64:server.example.com:enp1s0:none
ip=ip::gateway:netmask:hostname:interface:method:mtu	オーバーライドを使用した特定インターフェイスの自動設定

自動インターフェイスの設定方法

オーバーライドを使用した**特定インターフェイスの自動設定**では、**dhcp** など、指定した自動設定方法を使用してインターフェイスを起動しますが、自動取得した IP アドレス、ゲートウェイ、ネットマスク、ホスト名、他のパラメーターなどで指定したものは無効にします。パラ

メーターはすべて任意となるため、無効にするパラメーターだけを指定します。

method パラメーターには、以下のいずれかを使用します。

DHCP

dhcp

IPv6 DHCP

dhcp6

IPv6 自動設定

auto6

iBFT (iSCSI Boot Firmware Table)

ibft



注記

- **ip** オプションを指定せずに、**inst.ks=http://host/path** などのネットワークアクセスを必要とするブートオプションを使用する場合、**ip** オプションのデフォルト値は **ip=dhcp** です。
- iSCSI ターゲットに自動的に接続するには、**ip=ibft** ブートオプションを使用して、ターゲットにアクセスするネットワークデバイスをアクティブ化します。

nameserver=

nameserver= オプションは、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す。



注記

ip= パラメーターには角括弧が必要です。ただし、IPv6 アドレスには角括弧が使用できません。IPv6 アドレスに使用する正しい構文は **nameserver=2001:db8::1** のようになります。

bootdev=

bootdev= オプションは、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** オプションを複数回使用する場合に必要になります。

ifname=

ifname= オプションは、特定の MAC アドレスを持つネットワークデバイスにインターフェイス名を割り当てます。このオプションは複数回使用できます。構文は、**ifname=interface:MAC** です。以下に例を示します。

```
ifname=eth0:01:23:45:67:89:ab
```



注記

ifname= オプションは、インストール中にカスタムのネットワークインターフェイス名を設定する際にサポートされる唯一の方法となります。

inst.dhcpclass=

inst.dhcpclass= オプションは、DHCP のベンダークラス識別子を指定します。**dhcpcd** サービスではこの値を **vendor-class-identifier** として認識します。デフォルト値は **anaconda-\$(uname -srm)** です。

inst.waitfornet=

inst.waitfornet=SECONDS 起動オプションを使用すると、インストールシステムは、ネットワーク接続を待ってからインストールします。**SECONDS** 引数で指定する値は、ネットワーク接続がない場合でもすぐにはタイムアウトにせず、ネットワーク接続を待ち続け、インストールプロセスを継続する最大秒数を表します。

vlan=

vlan= オプションを使用して、仮想 LAN (VLAN) デバイスに特定の名前を付け、指定インターフェイスにそのデバイスを設定します。構文は **vlan=name:interface** です。以下に例を示します。

```
vlan=vlan5:enp0s1
```

これにより、**enp0s1** インターフェイスに **vlan5** という名前の VLAN デバイスが設定されます。name は以下のような形式をとります。

- VLAN_PLUS_VID: **vlan0005**
- VLAN_PLUS_VID_NO_PAD: **vlan5**
- DEV_PLUS_VID: **enp0s1.0005**
- DEV_PLUS_VID_NO_PAD: **enp0s1.5**

bond=

bond= オプションを使用して、**bond=name[:interfaces][:options]** 構文でボンディングデバイスを設定します。**name** はボンディングデバイス名に置き換え、**interfaces** は物理 (イーサネット) インターフェイスのコンマ区切りリストに置き換え、**options** はボンディングオプションのコンマ区切りリストに置き換えます。以下に例を示します。

```
bond=bond0:enp0s1,enp0s2:mode=active-backup,tx_queues=32,downdelay=5000
```

利用可能なオプションのリストは、ボンディングコマンド **modinfo** を実行します。

team=

team= オプションを使用して、**team=name:interfaces** 構文でチームデバイスを設定します。チームデバイスの基礎となるインターフェイスとして使用されるように、**name** はチームデバイスの望ましい名前に、**interfaces** は物理 (イーサネット) デバイスのコンマ区切りリストに置き換えます。以下に例を示します。

```
team=team0:enp0s1,enp0s2
```



重要

Red Hat Enterprise Linux 9 では、ネットワークチーミングが非推奨になりました。代わりに、ネットワークボンディングドライバーの使用を検討してください。詳細は、[Configuring network bonding](#) を参照してください。

bridge=

bridge= オプションを使用して、**bridge=name:interfaces** 構文でブリッジデバイスを設定します。ブリッジデバイスの基礎となるインターフェイスとして使用されるように、**name** はブリッジデバイスの望ましい名前に、**interfaces** は物理 (イーサネット) デバイスのコマ区切りリストに置き換えます。以下に例を示します。

```
bridge=bridge0:enp0s1,enp0s2
```

関連情報

- [ネットワークの設定および管理](#)

1.5. コンソール起動オプション

このセクションでは、コンソール、モニターディスプレイ、およびキーボードの起動オプションを設定する方法を説明します。

console=

console= オプションを使用して、プライマリーコンソールとして使用するデバイスを指定します。たとえば、最初のシリアルポートでコンソールを使用するには、**console=ttyS0** を使用します。**console=** 引数を使用する場合、インストールはテキスト UI から始まります。**console=** オプションを複数回使用する必要がある場合は、指定したすべてのコンソールにブートメッセージが表示されます。ただし、インストールプログラムは、最後に指定されたコンソールのみを使用します。たとえば、**console=ttyS0 console=ttyS1** と指定すると、インストールプログラムでは **ttyS1** が使用されます。

inst.lang=

inst.lang= オプションを使用して、インストール時に使用する言語を設定します。ロケールのリストを表示するには、コマンド **locale -a | grep _** または **localectl list-locales | grep _** コマンドを実行します。

inst.geoloc=

インストールプログラムで、地理位置情報の使用方法を設定するには、**inst.geoloc=** オプションを使用します。地理位置情報は、言語およびタイムゾーンの事前設定に使用され、**inst.geoloc=value** 構文を使用します。**value** には、以下のいずれかのパラメーターを使用します。

- 地理位置情報の無効化: **inst.geoloc=0**
- Fedora GeolP API (**inst.geoloc=provider_fedora_geoip**) の使用。このオプションは非推奨となりました。
- Hostip.info GeolP API (**inst.geoloc=provider_hostip**) の使用。このオプションは非推奨となりました。

inst.keymap=

inst.keymap= オプションを使用して、インストールに使用するキーボードレイアウトを指定します。

inst.cmdline

inst.cmdline オプションを使用して、インストールプログラムをコマンドラインモードで強制的に実行します。このモードでは対話ができないため、キックスタートファイルまたはコマンドラインですべてのオプションを指定する必要があります。

inst.graphical

インストールプログラムをグラフィカルモードで強制的に実行するには、**inst.graphical** オプションを使用します。グラフィカルモードがデフォルトです。

inst.text

inst.text オプションを使用して、グラフィカルモードではなく、テキストモードでインストールプログラムを強制的に実行します。

inst.noninteractive

inst.noninteractive 起動オプションを使用して、非対話モードでインストールプログラムを実行します。非対話型モード(および **inst.noninteractive**) では、ユーザーとの対話は許可されていません。グラフィカルまたはテキストインストールで **inst.noninteractive** オプションを使用できません。**inst.noninteractive** オプションをテキストモードで使用すると、**inst.cmdline** オプションと同じように動作します。

inst.resolution=

inst.resolution= オプションを使用して、グラフィカルモードで、画面の解像度を指定します。形式は **NxM** です。N は画面の幅で、M は画面の高さ (ピクセル単位) です。推奨される解像度は 1024x768 です。

inst.vnc

inst.vnc オプションを使用して、Virtual Network Computing (VNC) を使用したグラフィカルインストールを実行します。インストールプログラムと対話するには VNC クライアントアプリケーションを使用する必要があります。VNC 共有を有効にすると、複数のクライアントに接続できます。VNC を使用してインストールしたシステムは、テキストモードで起動します。

inst.vncpassword=

inst.vncpassword= オプションを使用して、インストールプログラムが使用する VNC サーバーにパスワードを設定します。

inst.vncconnect=

inst.vncconnect= オプションを使用して、指定されたホストの場所にあるリスニング VNC クライアントに接続します (例: **inst.vncconnect=<host>[:<port>]**)。デフォルトのポートは 5900 です。このオプションを使用するには、コマンド **vncviewer -listen** を入力します。

inst.xdriver=

inst.xdriver= オプションを使用して、インストール時およびインストール済みシステムで使用される X ドライバーの名前を指定します。

inst.usefbx

inst.usefbx オプションを使用して、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 ドライバーを使用するようにインストールプログラムに要求します。このオプションは、**inst.xdriver=fbdev** オプションと同等です。

modprobe.blacklist=

modprobe.blacklist= オプションを使用して、1つ以上のドライバーを拒否リストに追加するか、完全に無効にします。このオプションを使用して無効にしたドライバー (mods) は、インストールの開始時にロードできません。インストールが完了すると、インストールされたシステムはこれらの設定を保持します。拒否リストに指定したドライバーのリストは、**/etc/modprobe.d/** ディレクトリにあります。複数のドライバーを無効にするには、コンマ区切りリストを使用します。以下に例を示します。

```
modprobe.blacklist=ahci,firewire_ohci
```

注記

modprobe.blacklist は、さまざまなコマンドラインオプションと組み合わせて使用できます。たとえば、既存のドライバーの更新バージョンがドライバー更新ディスクから確実に読み込まれるようにするには、**inst.dd** オプションを使用します。

```
modprobe.blacklist=virtio_blk
```

inst.xtimeout=

inst.xtimeout= オプションを使用して、X サーバーの起動のタイムアウトを秒単位で指定します。

inst.sshd

インストール時に、SSH を使用してシステムに接続し、インストールの進捗を監視できるように、**inst.sshd** オプションを使用して、**sshd** サービスを開始します。SSH の詳細は、man ページの **ssh(1)** を参照してください。デフォルトでは、**sshd** オプションは、64 ビットの IBM Z アーキテクチャーでのみ自動的に起動します。その他のアーキテクチャーでは、**sshd** は、**inst.sshd** オプションを使用しない限り起動しません。

**注記**

インストール中に、root アカウントにはデフォルトでパスワードが設定されていません。キックスタートコマンド **sshpw** を使用して、インストール時に root パスワードを設定できます。

inst.kdump_addon=

インストールプログラムで Kdump 設定画面 (アドオン) を有効または無効にするには、**inst.kdump_addon=** オプションを使用します。この画面はデフォルトで有効になっているため、無効にする場合は **inst.kdump_addon=off** を使用します。アドオンを無効にすると、グラフィカルおよびテキストベースのインターフェイスと、キックスタートコマンド **%addon com_redhat_kdump** の両方で Kdump 画面が無効になります。

1.6. 起動オプションのデバッグ

このセクションでは、問題をデバッグするときに使用できるオプションを説明します。

inst.rescue

inst.rescue オプションを使用して、システムの診断と修正のためのレスキュー環境を実行します。たとえば、[レスキューモードでファイルシステムを修復](#) できます。

inst.updates=

inst.updates= オプションを使用して、インストール時に適用する **updates.img** ファイルの場所を指定します。**updated.img** ファイルは、いくつかのソースの1つから取得できます。

表1.4 **updates.img** ファイルソース

ソース	説明	例
ネットワークからの更新	updates.img のネットワーク上の場所を指定します。インストールツリーを変更する必要はありません。この方法を使用するには、カーネルコマンドラインを編集して inst.updates を追加します。	inst.updates=http://website.com/path/to/updates.img.

ソース	説明	例
ディスクイメージからの更新	フロッピードライブまたは USB キーに updates.img を保存できます。これは、ファイルシステムタイプが ext2 の updates.img でのみ可能です。イメージの内容をフロッピードライブに保存するには、フロッピーディスクを挿入し、次のコマンドを実行します。	dd if=updates.img of=/dev/fd0 bs=72k count=20 USB キーまたはフラッシュメディアを使用するには、 /dev/fd0 を、USB キーのデバイス名に置き換えます。
インストールツリーからの更新	CD、ディスク、HTTP、または FTP のインストールを使用する場合は、すべてのインストールツリーが .img ファイルを検出できるように、インストールツリーに updates.img を保存できます。このファイル名は、 updates.img にする必要があります。	NFS インストールの場合は、ファイルを images/ ディレクトリまたは RHupdates/ ディレクトリに保存します。

inst.syslog=

インストールの開始時に、指定されたホスト上の **syslog** プロセスにログメッセージを送信します。**inst.syslog=** は、リモート **syslog** プロセスが着信接続を受け入れるように設定されている場合にのみ使用できます。

inst.virtioiog=

inst.virtioiog = オプションを使用して、ログの転送に使用する virtio ポート (**/dev/virtio-ports/name** にある文字デバイス) を指定します。デフォルト値は、**org.fedoraproject.anaconda.log.0** です。

rd.live.ram

images/install.img の **stage 2** イメージを RAM にコピーします。これにより、インストールに必要なメモリーがイメージのサイズ (通常は 400 ~ 800MB) だけ増加することに注意してください。

inst.nokill

致命的なエラーが発生したとき、またはインストールプロセスの最後に、インストールプログラムが再起動しないようにします。再起動時に失われるインストールログをキャプチャーするのに使用します。

inst.noshell

インストール中にターミナルセッション 2 (tty2) でシェルを防止します。

inst.notmux

インストール中に tmux を使用しないようにします。この出力は、ターミナル制御文字なしで生成され、非対話用になります。

inst.remotelog=

TCP 接続を使用してすべてのログをリモート **host:port** に送信します。リスナーがなく、インストールが通常通りに進まない場合は、接続が中断されます。

1.7. ストレージ起動オプション

このセクションでは、ストレージデバイスからの起動をカスタマイズするために指定できるオプションを説明します。

inst.nodmraid

dmraid サポートを無効にします。



警告

使用する場合は注意が必要です。ファームウェア RAID アレイの一部として誤って特定されたディスクがある場合は、古い RAID メタデータが存在する可能性があります。これらは、**dmraid** や **wipefs** などの適切なツールを使用して削除する必要があります。

inst.nompath

マルチパスデバイスのサポートを無効にします。このオプションは、システムに誤検知があり、通常のブロックデバイスをマルチパスデバイスとして誤って識別する場合にのみ使用してください。



警告

使用する場合は注意が必要です。マルチパスハードウェアではこのオプションを使用しないでください。このオプションを使用してマルチパスデバイスのシングルパスにインストールすることはサポートされていません。

inst.gpt

インストールプログラムがパーティション情報を Master Boot Record (MBR) ではなく GUID Partition Table (GPT) にインストールするように強制します。このオプションは、BIOS 互換モードである場合を除き、UEFI ベースのシステムでは有効ではありません。通常、BIOS 互換モードの BIOS ベースのシステムおよび UEFI ベースのシステムは、ディスクのサイズが 2^{32} セクター以上でない限り、パーティション情報の格納に MBR スキーマを使用しようとします。ディスクセクターは通常 512 バイトで、通常これは 2 TiB に相当します。**inst.gpt** ブートオプションを使用すると、GPT をより小さなディスクに書き込むことができます。

inst.wait_for_disks=

inst.wait_for_disks= オプションを使用して、インストールの開始時にディスクデバイスが表示されるまでインストールプログラムが待機する秒数を指定します。キックスタートファイルまたはカーネルドライバを自動的にロードするために **OEMDRV-labeled** デバイスを使用しているものの、起動プロセス中にデバイスが表示されるまでに時間がかかる場合は、このオプションを使用します。デフォルトでは、インストールプログラムは **5** 秒間待機します。遅延を最小限に抑えるには、**0** 秒を使用します。

1.8. キックスタート起動オプション

このセクションでは、インストールを自動化するのにキックスタートファイルに追加できるブートオプションを説明します。

inst.ks=

インストールの自動化に使用するキックスタートファイルの場所を定義します。その後、いずれかの **inst.repo** 形式を使用して、場所を指定できます。パスを指定せずにデバイスを指定すると、インストールプログラムは、指定したデバイスの **/ks.cfg** でキックスタートファイルを検索します。

デバイスを指定せずにこのオプションを使用する場合、インストールプログラムはオプションに次の値を使用します。

```
inst.ks=nfs:next-server:/filename
```

ここでは、**next-server** は DHCP の **next-server** オプション、または DHCP サーバーの IP アドレスで、**filename** は DHCP の **filename** オプションまたは **/kickstart/** です。指定のファイル名が / 文字で終了すると、**ip-kickstart** が追加されます。次の表に例を示します。

表1.5 デフォルトのキックスタートファイルの場所

DHCP サーバーのアドレス	クライアントのアドレス	キックスタートファイルの場所
192.168.122.1	192.168.122.100	192.168.122.1:/kickstart/192.168.122.100-kickstart

OEMDRV のラベルが付いたボリュームが存在すると、インストールプログラムは、キックスタートファイル **ks.cfg** を読み込もうとします。キックスタートファイルがこの場所にある場合は、**inst.ks=** 起動オプションを使用する必要がありません。

inst.ks.all

複数の **inst.ks** オプションによる複数のキックスタートファイルの場所を順次試行するように **inst.ks.all** オプションを指定します。最初に成功した場所が使用されます。これは、**http**、**https**、または **ftp** タイプの場所のみ適用され、その他の場所は無視されます。

inst.ks.sendmac

inst.ks.sendmac オプションを使用して、すべてのネットワークインターフェイスの MAC アドレスを含む HTTP 送信リクエストにヘッダーを追加します。以下に例を示します。

```
X-RHN-Provisioning-MAC-0: eth0 01:23:45:67:89:ab
```

これは、**inst.ks=http** を使用してシステムをプロビジョニングする場合に便利です。

inst.ks.sendsn

inst.ks.sendsn オプションを使用して、HTTP 送信リクエストにヘッダーを追加します。このヘッダーには、**/sys/class/dmi/id/product_serial** から読み込まれたシステムのシリアル番号が含まれます。ヘッダーの構文は以下のとおりです。

```
X-System-Serial-Number: R8VA23D
```

関連情報

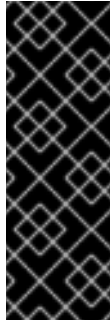
- [Full list of boot options](#)

1.9. 高度なインストール起動オプション

本セクションでは、高度なインストール起動オプションを説明します。

inst.kexec

再起動を実行する代わりに、インストールの最後に **kexec** システムコールを実行します。**inst.kexec** オプションは、新しいシステムを即座に読み込み、BIOS またはファームウェアが通常実行するハードウェアの初期化を回避します。



重要

このオプションは非推奨になっており、テクノロジープレビューとしてのみ利用できません。テクノロジープレビュー機能に対する Red Hat のサポート範囲の詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

kexec を使用すると、通常はシステムの完全な再起動時にクリアされるデバイスレジスタがデータでいっぱいになる可能性があります。これにより、特定のデバイスドライバーに問題が発生する可能性があります。

inst.multilib

multilib パッケージ用にシステムを設定して、64 ビット AMD64 または Intel 64 システムに 32 ビットパッケージをインストールできるようにします。通常、AMD64 または Intel 64 システムでは、このアーキテクチャー専用となるパッケージ (x86_64 の印が付いている) と、全アーキテクチャー用のパッケージ (noarch の印が付いている) がインストールされます。**inst.multilib** 起動オプションを使用すると、32 ビットの AMD または Intel システム用のパッケージ (i686 の印が付いている) が自動的にインストールされます。

これは、**%packages** セクションで直接指定されているパッケージにのみ適用されます。パッケージが依存関係としてインストールされている場合は、正確に指定した依存関係のみがインストールされます。たとえば、**glibc** パッケージに依存する **bash** パッケージをインストールする場合、**bash** パッケージは複数のバリエーションでインストールされますが、**glibc** パッケージは **bash** パッケージが必要とするバリエーションにのみインストールされます。

selinux=0

インストールプログラムおよびインストールされたシステムでの SELinux の使用を無効にします。デフォルトでは、SELinux はインストールプログラムでは permissive モードで動作し、インストールされたシステムでは enforcing モードで動作します。



注記

inst.selinux=0 と **selinux=0** のオプションは同じではありません。* **inst.selinux=0**: は、インストールプログラムでのみ SELinux を無効にします。* **selinux=0**: インストールプログラムおよびインストールされたシステムでの SELinux の使用を無効にします。SELinux を無効にすると、イベントがログに記録されなくなります。

inst.nonibftiscsiboot

iSCSI ブートファームウェアテーブル (iBFT) で設定されていない iSCSI デバイスにブートローダーを配置します。

1.10. 削除済みの起動オプション

以下の起動オプションは、Red Hat Enterprise Linux から削除されました。

inst.zram

zram.service はこれ以上実行できません。詳細は、**zram-generator** を参照してください。

inst.singlelang

シングル言語モードはサポートされなくなりました。

inst.loglevel

ログレベルは常に debug に設定されます。