

EulerOS V2.0SP5 升级指导书

文档版本 01
发布日期 2019-08-12



版权所有 © 华为技术有限公司 2019。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址：深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址：<http://www.huawei.com>

客户服务邮箱：support@huawei.com

客户服务电话：4008302118

目 录

1 yum 升级场景	1
1.1 升级说明	1
1.1.1 注意事项	1
1.1.2 同版本升级	1
1.1.3 跨版本升级	1
1.2 升级步骤	2
1.2.1 配置更新 yum repo	2
1.2.2 规避可能出现的问题	2
1.2.3 开始升级所有的包和他们的依赖	2
1.2.4 取消规避操作	2
1.2.5 重启系统	2
1.3 配置升级 repo	2
1.3.1 获取 ISO 镜像	2
1.3.2 挂载 ISO 并配置为 repo 源	3
1.4 升级问题说明	4
1.4.1 升级前可能存在的冲突	4
1.4.1.1 冲突一	4
1.4.1.1.1 原因	4
1.4.1.1.2 解决方案	4
1.4.1.2 冲突二	5
1.4.1.2.1 原因	5
1.4.1.2.2 解决方法	5
1.4.1.3 冲突三	5
1.4.1.3.1 原因	6
1.4.1.3.2 解决方法	6
1.4.2 升级中可能出现的问题	6
1.4.2.1 解决方法	6
1.4.3 不支持 yum 方式升级的 rpm 包	6
2 双区升级场景（for ELB 弹性负载均衡）	7
2.1 双区升级前检查	7
2.2 双区升级场景	9

1 yum 升级场景

[1.1 升级说明](#)

[1.2 升级步骤](#)

[1.3 配置升级repo](#)

[1.4 升级问题说明](#)

1.1 升级说明

yum update可以检查您的系统中是否有软件包需要更新。您可以通过yum一次性全部更新或者选择对应的包单独更新。

1.1.1 注意事项

1. 在更新中，突然掉电或更新进程被强制中止，会有系统无法重新启动的风险。
2. 请保证根分区有足够的存储空间，一般要求大于8G，否则升级可能会失败。
3. 升级后不支持进入原内核启动项，因为在新内核中路由信息接口有变更，涉及到monitor_netdev模块，在原来内核启动中会出现oops。
4. 升级成功后需要重启系统。

1.1.2 同版本升级

现已支持EulerOS 2.5 V200R007C00B603同版本升级所有包到EulerOS 2.5 V200R007最新版本。

1.1.3 跨版本升级

- 现已支持EulerOS 2.2.16跨版本升级所有包到EulerOS 2.5 V200R007C00B603版本。
- 暂不支持EulerOS 2.2.16跨版本升级到EulerOS 2.5 V200R007最新版本。
- 如果从EulerOS 2.2.16跨版本升级到EulerOS 2.5 V200R007最新版本，可以先跨版本过渡升级到V200R007C00B603，再同版本升级到EulerOS V200R007最新版本。

1.2 升级步骤

无论是跨版本还是同版本通过yum update升级所有包，基本的升级流程都是一样的。

1.2.1 配置更新 yum repo

升级前需要配置需要升级的repo，可以按文档[1.3 配置升级repo](#)章节内容，配置更新yum repo。

1.2.2 规避可能出现的问题

在升级前规避升级中可能出现的一直停留在某一阶段的问题，需要注销proc-sys-fs-binfmt_misc.automount服务。

```
# systemctl mask proc-sys-fs-binfmt_misc.automount
```

1.2.3 开始升级所有的包和他们的依赖

执行yum update后，如果是跨版本升级，可能会报有冲突，可按文档[1.4 升级问题说明](#)章节处理。

```
# yum update
```

1.2.4 取消规避操作

在升级后取消规避操作，取消注销proc-sys-fs-binfmt_misc.automount服务。

```
# systemctl unmask proc-sys-fs-binfmt_misc.automount
```

1.2.5 重启系统

升级完成后，重启系统。

```
# reboot
```

1.3 配置升级 repo

升级EulerOS前需要配置yum repo，以下为配置本地yum repo方法。

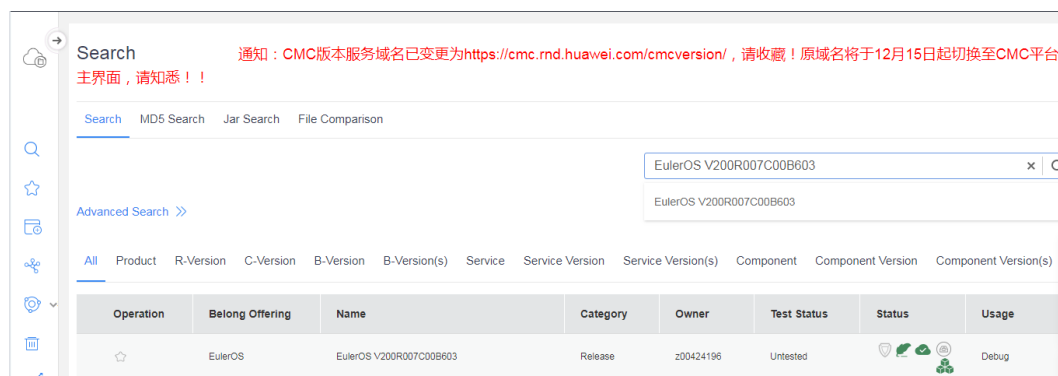
1.3.1 获取 ISO 镜像

通过CMC上获取EulerOS 2.5 V200R007版本的ISO。

步骤1 登录CMC，网址为：<https://cmc.rnd.huawei.com/cmceversion/>。

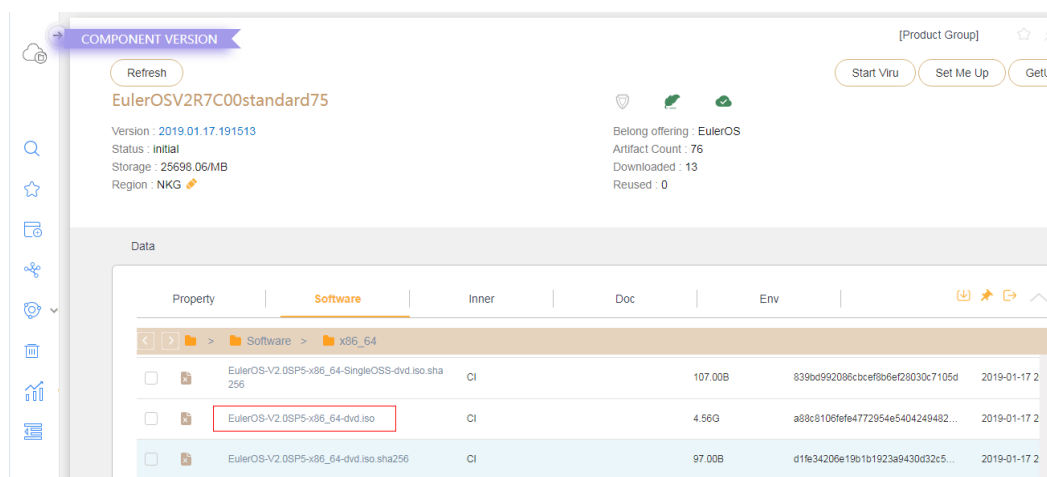
步骤2 查找需要升级的EulerOS_V200R007xxx版本，以“EulerOS V200R007C00B603”为例。

图 1-1 从 CMC 上获取 ISO



步骤3 点击发布包名称，在Software相应目录下获取EulerOS-V2.0SP5-x86_64-dvd.iso。

图 1-2 获取发布包



说明

获取发布包后，请进行完整性校验，建议您使用PGP数字签名进行验证，详细的验证操作说明请参见“安全资料—发布包完整性校验”章节。

----结束

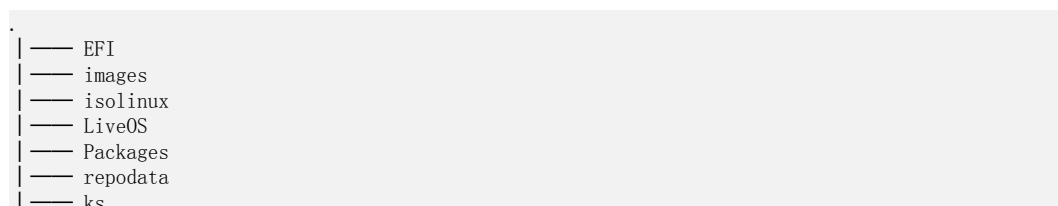
1.3.2 挂载 ISO 并配置为 repo 源

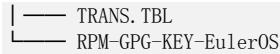
使用mount命令挂载镜像文件。

示例如下：

```
# mount /home/Euler/EulerOS-V2.0SP5-x86_64-dvd.iso /mnt/
```

挂载好的mnt目录如下：





```
graph LR
    TRANS_TBL[TRANS. TBL] --- RPM_GPG_KEY[RPM-GPG-KEY-EulerOS]
```

其中，`Packages`为rpm包所在的目录，`repodata`为repo源元数据所在的目录，`RPM-GPG-KEY-EulerOS`为EulerOS的签名公钥。

挂载后的目录可以配置为yum源使用，在`/etc/yum.repos.d/`目录下创建`***.repo`的配置文件（必须以`.repo`为扩展名）。

示例如下：

在`/etc/yum.repos.d`目录下创建 `EulerOS-2.5-Base.repo` 文件，使用本地镜像挂载目录作为yum源，`EulerOS-2.5-Base.repo`的内容如下：

```
[base]
name=base
baseurl=file:///mnt
enabled=1
gpgcheck=1
gpgkey=file:///mnt/RPM-GPG-KEY-EulerOS
```

说明

- `gpgcheck`可设置为1或0，1表示进行gpg（GNU Private Guard）校验，0表示不进行gpg校验，`gpgcheck`可以确定rpm包的来源是有效和安全的。
- `gpgkey`为签名公钥的存放路径。

1.4 升级问题说明

EulerOS 2.2.16通过yum update跨版本更新所有的包到EulerOS 2.5 V200R007C00B603时，可能存在一些问题。

1.4.1 升级前可能存在的冲突

EulerOS 2.2.16通过yum update更新所有的包到EulerOS 2.5 V200R007C00B603时，可能会报有冲突。

1.4.1.1 冲突一

执行 `yum update`后，yum会分析出如下错误：

```
Error: python2-ipalib conflicts with ipa-python.x86_64
Error: ipa-common conflicts with ipa-python.x86_64
```

1.4.1.1.1 原因

`ipa-python.x86_64`已经在EulerOS 2.5版本中移除了，被`python2-ipalib`和`ipa-common`这两个包代替了，但是升级的的repo中没有`ipa-python.x86_64`对应的rpm升级包`ipa-python-compatible.noarch`，因此造成了冲突。

1.4.1.1.2 解决方案

通过`yum remove`卸载`python2-ipalib`及其依赖的rpm，其中依赖的rpm如果需要可以通过配置后的升级yum源重新安装新版本。

示例如下：

1. 通过yum update尝试更新所有的包，yum报错。

```
# yum update
Resolving Dependencies
...
--> Package mythes-en.noarch 0:3.0-13.eulerosv2r7 will be installed
--> Package rasqal.x86_64 0:0.9.30-4.eulerosv2r7 will be installed
--> Processing Conflict: python2-ipalib-4.5.4-10.3.h2.eulerosv2r7.noarch conflicts ipa-
python < 4.2.91
--> Processing Conflict: ipa-common-4.5.4-10.3.h2.eulerosv2r7.noarch conflicts ipa-python <
4.2.91
--> Finished Dependency Resolution
Error: python2-ipalib conflicts with ipa-python-4.2.0-15.0.1.19.h4.x86_64
Error: ipa-common conflicts with ipa-python-4.2.0-15.0.1.19.h4.x86_64
You could try using --skip-broken to work around the problem
You could try running: rpm -Va --nofiles -nodigest
```

2. 通过yum remove卸载ipa-python。

```
# yum remove ipa-python
Resolving Dependencies
...
Dependencies Resolved

=====
Package                                Arch                                Version
=====
Removing:
ipa-python                             x86_64                             4.2.0-15.0.1.19.h4
Removing for dependencies:
ipa-admintools                         x86_64                             4.2.0-15.0.1.19.h4
ipa-client                             x86_64                             4.2.0-15.0.1.19.h4
ipa-server                             x86_64                             4.2.0-15.0.1.19.h4

Transaction Summary
=====
Remove 1 Package (+3 Dependent packages)

Installed size: 13 M
Is this ok [y/N]:
```

3. 通过配置好的升级repo，重新安装依赖包，ipa-client，ipa-server ipa-admintools。

```
# yum install ipa-client ipa-server ipa-admintools
```

1.4.1.2 冲突二

执行yum update后，yum会分析出如下错误：

```
Error: Protected multilib versions: libcgrouper.x86_64 != libcgrouper.i686
```

1.4.1.2.1 原因

系统中同时安装了相同包的不同架构的包libcgrouper.x86_64和libcgrouper.i686，造成了yum冲突。

1.4.1.2.2 解决方法

通过yum remove卸载libcgrouper.i686，如果需要，可在yum update更新全部rpm包后单独重新安装。

1.4.1.3 冲突三

执行yum update后，yum会分析出如下错误：

```
Error: Protected multilib versions: libvirt-client.x86_64 != libvirt-client.i686
```

1.4.1.3.1 原因

系统中同时安装了相同包的不同架构的包libvirt-client.x86_64和libvirt-client.i686，造成了yum冲突。

1.4.1.3.2 解决方法

通过yum remove卸载libvirt-client.i686，如果需要，可在yum update更新全部rpm包后单独重新安装。

1.4.2 升级中可能出现的问题

通过yum update更新所有的包到EulerOS 2.5 V200R007C00B603时，安装过程中systemd会多次reload服务，可能会引起systemd关于automount的bug，现象为安装进程会一直停留在安装某一rpm阶段。

1.4.2.1 解决方法

在升级前注销服务：

```
# systemctl mask proc-sys-fs-binfmt_misc.automount
```

在升级后取消注销服务：

```
# systemctl unmask proc-sys-fs-binfmt_misc.automount
```

说明

如果出现此问题，可切换到系统的其他终端中，执行如下命令：

```
# systemctl restart proc-sys-fs-binfmt_misc.automount
```

1.4.3 不支持 yum 方式升级的 rpm 包

在EulerOS 2.2.16升级EulerOS 2.5 V200R007C00B603过程中，EulerOS 2.2.16批量安装ISO中所有的RPM时，部分RPM之间会产生冲突，或缺少部分库文件无法安装，这些问题会中断安装，所以在全量安装RPM包时过滤了部分会造成安装中断的RPM包。

如下这些RPM包同时安装时会造成安装中断，需要在安装时过滤不进行安装，所以这些RPM也不支持同时升级到EulerOS 2.5 V200R007C00B603版本。

1. rsyslog8.x86_64：会和rsyslog.x86_64冲突。
2. libcmpiCppImpl0.x86_64：会和tog-pegasus-libs.i686，tog-pegasus-libs.x86_64冲突。
3. sssd-common.i686：会和sssd-common.x86_64冲突。
4. libtiff-devel.i686：会和libtiff-devel.x86_64冲突。
5. sssd-krb5-common.i686：缺少部分库文件，无法安装。
6. qat.x86_64：会和linux-firmware.noarch冲突。
7. qat-engine.x86_64：会和linux-firmware.noarch和qat.x86_64冲突。
8. qat-devel.x86_64：会和linux-firmware.noarch和qat.x86_64冲突。
9. oxygen-gtk.noarch：会和oxygen-gtk3.x86_64冲突。

说明

批量安装时部分RPM之间会产生冲突，用户如需安装，请单独进行升级处理。

2 双区升级场景（for ELB 弹性负载均衡）

本特性支持ELB弹性负载均衡产品。

2.1 双区升级前检查

2.2 双区升级场景

2.1 双区升级前检查

注意事项

1. 升级前检查脚本不支持并发执行，同一时刻只能有一个升级程序运行。
2. 准备备区，无预留分区的产品，且使用了逻辑分区，可以通过逻辑分区扩容的方式调整分区大小，再新增一个备区。

说明

- 准备备区由产品实现。
 - 扩容有风险，请产品做好备份，极端情况需要重新安装。
3. 新建/os2目录，并把新增备区挂载到os2,修改/etc/fstab，添加“/dev/sdc /os2 ext4 defaults 1 2”，其中/dev/sdc需修改为实际备区盘。
 4. 升级前检查日志为/var/log/euler_patch.log。
 5. 请确保/os2、根分区、/boot分区挂载正常，否则检查失败。
 6. 请保证/boot分区中关键文件，如grub.cfg正常，否则检查失败。
 7. 内存盘场景下请保证根分区剩余空间大于20M；非内存盘场景请保证根分区剩余空间大于200M；否则检查失败。
 8. 在/var/log单独挂载情况下，请保证/var/log所在分区使用率需小于90%，否则检测失败。
 9. 首次升级时/boot/efi/EFI/euleros/grub.cfg中存在多个引导项，可能导致升级前检查成功，但升级失败。
 10. 如果系统为uefi启动，需修改配置文件/opt/osupgrade/bin/euleros-upgrade.conf，将EULEROS_EFI_BOOT_CONF配成对应grub.cfg位置，该位置是相对于/boot/efi的一个相对位置，如/boot/efi/grub2/grub.cfg，EULEROS_EFI_BOOT_CONF需配成grub2/grub.cfg。
 11. 升级前后版本的grub.cfg文件中，内核和initrd启动命令前不可加调试信息，如time等。

```

echo 'List boot files...'
ls -l /boot/
echo 'Loading Linux kernel ...'
time linux /boot/bzImage rdinit=/sc_init user_root=tmpfs rw add_efi_memmap console=tty0 console=ttyS0,115200 earlyprintk=ttyS0,115200 elevator
=nop nohop pci=hpioirqs=0,hmmmsize=0 ide=nodma apic acpi_wake_gpes_always_on no_8500 mce pbs_size=128M loglevel=8 oops=panic panic=3 softlockup_pa
nicol reserve_kbox_mem=16M audit=1 bootfs=ext4 net.ifnames=0 nohz=off intel_idle.max_cstate=0 selinux=0 intel_pstate=disable nopath erst_disable rd.shel
l=0 console=tty0 console=ttyS0,115200
## echo 'Loading devicetree ...'
devicetree /boot/conf/hisi_p660_evb_pangea_for_grub.dtb
echo 'Loading initial ramdisk ...'
time initrd /boot/initrd
echo 'Loading initial ramdisk ...ok'

```

升级前检查流程

升级前检查脚本是否已安装在当前环境中，若当前环境中没有升级脚本可以通过安装rpm包的方式获取。

执行升级相关操作前，需修改配置文件/opt/osupgrade/bin/euleros-upgrade.conf，将EULEROS_BACKUP_PARTITION_SIZE_NO_CHECK配置成true，关闭备区大小检查，备区大小由产品自己确保。

- 执行升级前检查

```
# sh /opt/osupgrade/bin/install_patch.sh -q
```

- 查看返回值

```
# echo $?
```

返回0表示检查成功，其他表示检查失败。

接口说明

- 升级接口：
install_patch.sh -q
- 参数说明：
-q： 双驱升级前检查。
- 返回值：
0： 成功。
其他： 失败，详细情况见[表2-1](#)。

错误码

表 2-1 错误码

错误码	对应的信息说明
0	执行正常
1	执行失败（不属于下面几种原因的都返回1）
2	文件未找到
3	空间不足
4	获取挂载点失败
5	备份文件失败
7	未找到根目录
8	并发调用补丁脚本失败

错误码	对应的信息说明
9	参数错误
10	捕获信号异常中止
11	无法找到命令
13	文件类型错误
14	根文件系统错误
15	文件检查错误
16	根分区跟备分区大小不一致
17	无法找到磁盘
24	用户权限错误
25	日志分区检查错误
53	/boot/与当前环境不匹配

2.2 双区升级场景

注意事项

1. 升级脚本不支持并发执行，同一时刻只能有一个升级程序运行。
2. 升级包包含欧拉相关更新，不负责其他产品组件升级。其他产品组件升级可利用升级包本身的hook机制联合升级或者单独进行升级。
3. 升级时某些服务需要关闭，如停止业务的部分进程，监控。
4. 在升级过程中，在操作boot目录时，突然掉电或升级程序被强制中止，会有系统无法启动的风险。
5. 升级失败回滚功能不支持异常终止（例如掉电，重启等）导致的失败，但可继续升级。
6. 升级过程日志文件为/var/log/euler_patch.log。
7. 在/var/log单独挂载情况下，请确保磁盘剩余空间大于10%。
8. 请保证根分区剩余空间大于200M，否则会导致升级检查失败。
9. 内存盘场景下请确保内存盘剩余空间大于20M。
10. 产品需保证备区大小能容纳升级后的系统。
11. 请勿将全量包放置于/boot或者/os2目录，否则可能会导致系统升级成功后重启失败，或升级失败。
12. 仅支持回退一次，即基线版本，先打补丁A，再打补丁B，此时，从B仅能回退到补丁A的状态，不能回退到基线版本。
13. 升级及回退均需要重启才能切换系统。
14. 升级过程中使用kill -9中断偶尔会导致备区挂载的dev等目录无法卸载（例如升级脚本在执行chroot时），需要重启系统才可继续升级。
15. 执行回退流程前，确保/var/log、/os2、/boot分区挂载正常，否则回退会失败。

16. 升级操作不对boot分区完整性进行检查，boot分区部分文件缺失不影响升级但可能造成系统无法回退。
17. 回退过程被中断将无法恢复到回退前的场景，虽然被中断但可能已经回退成功。
18. 内存盘场景升级请务必保证有足够的可用内存，否则可能导致oom，建议最小值为300M。
19. /boot目录下的文件、目录不支持加保护，否则会导致回退时失败。
20. ISO中/boot/efi/EFI/euleros/grub.cfg已经存在的场景，升级脚本直接使用该grub文件，不再继承当前环境的grub配置。
21. 升级前后版本的grub.cfg文件中，内核和initrd启动命令前不可加调试信息，如time等。

```

echo 'List boot files...'
ls -l /boot/
echo 'Loading Linux kernel ...'
time linux /boot/bzImage rdinit=/sc_init user_root=tmpfs rw add_efi_memmap console=tty0 console=ttyS0,115200 earlyprintk=ttyS0,115200 elevator
=noop showopts pci=hpbiosize=0,hpmemsize=0 ide=nodma apic acpi_wake_gpes_always on no_8508 mce pbs_size=128M loglevel=8 oops=panic panic=3 softlockup_pa
nic=1 reserve_kbox_mem=16M audit=1 bootfs=ext4 net.ifnames=0 nohz=off intel_idle.max_cstate=0 selinux=0 intel_pstate=disable nopath erst_disable rd.shel
l=0 console=tty0 console=ttyS0,115200
### echo 'Loading device tree ...'
devicetree /boot/conf/hisi_p660_evb_pangea_for_grub.dtb
echo 'Loading initial ramdisk ...'
time initrd /boot/initrd
echo 'Loading initial ramdisk ...ok'

```

升级双启动项说明

1. 升级支持双启动项，升级后可以从升级版本或者基线版本启动，默认第一启动项为升级版本；
2. 在升级版本出现启动失败情况下，可以从第二启动项启动，进行升级回退操作，回退后需要重启，恢复到升级前状态；
3. 支持双启动项增加了/boot/bak文件夹和grub2/目录下的grub.cfg.euler_bak文件，用户不可以修改或删除，否则可能启动失败。
4. 从高版本执行升级至不支持双区启动的低版本时，若grub2目录下存在配置文件grub.cfg.euler_bak需要重命名为grub.cfg，需要覆盖原有文件，之后再执行升级操作。

升级流程

说明

以下以全量ISO放于/data目录下，且升级操作在/data目录下进行为例。

步骤1 获取全量ISO

ELB需要对自己原本通用镜像iso进行重新打包操作，以满足双区升级对iso目录结构及文件的需求。

步骤2 升级

说明

- 升级脚本已安装在当前环境中（若当前环境中没有升级脚本可以通过安装rpm包的方式获取）。
- 执行升级相关操作前，需修改配置文件/opt/osupgrade/bin/euleros-upgrade.conf，将EULEROS_BACKUP_PARTITION_SIZE_NO_CHECK配置成true，关闭备区大小检查，备区大小由产品自己确保。

执行升级

```
# sh /opt/osupgrade/bin/install_patch.sh -u -p /data/EulerOS.x86_64.iso
```

说明

当前系统版本号与升级后的版本号且编译时间一致时可以通过指定-f参数进行强制升级。

查看返回值

```
# echo $?
```

返回0表示升级成功，其他表示升级失败。

步骤3 重启系统，使升级生效

重启系统。

----结束

说明

升级后可以从升级版本或老版本启动，默认第一启动选项为升级版本；第二启动选项为老版本。

回退流程

步骤1 回退前检查

```
# sh /opt/osupgrade/bin/install_patch.sh -c
```

说明

返回0表示可以回退，返回51表示不可以回退，其他表示接口执行失败。

查看返回值

```
# echo $?
```

返回0表示可以回退，返回51表示不可以回退，其他表示接口执行失败。

步骤2 回退

执行回退

```
# sh /opt/osupgrade/bin/install_patch.sh -r
```

说明

返回0表示回退成功，返回其他表示回退失败。

查看返回值

```
# echo $?
```

返回0表示回退成功，返回其他表示回退失败。

步骤3 重启系统，使回退生效

重启系统。

----结束

说明

升级后从第二启动选项启动老版本，状态为执行升级操作后状态，需要执行回退操作后重启生效。

接口说明

- 升级接口：

install_patch.sh -u [-f] -p <package>

参数说明：

-f: 该参数在系统版本号且编译时间一致的情况下依然进行强制升级。

-p: package指定了全量iso的绝对路径。

返回值:

0: 成功。

其他: 失败, 详细情况见[错误码](#)章节。

- 回退接口:

install_patch.sh -r

返回值:

0: 成功

其他: 失败, 详细情况见[错误码](#)章节。

- 回退检查接口:

install_patch.sh -c

返回值:

0: 可回退

51: 无需回退

其他: 失败, 详细情况见[错误码](#)章节。

错误码

错误码	对应的信息说明
0	执行正常
1	执行失败（不属于下面几种原因的都返回1）
2	文件未找到
3	空间不足
4	获取挂载点失败
5	备份文件失败
6	切换备区失败
7	未找到根目录
8	并发调用补丁脚本失败
9	参数错误
10	捕获信号异常中止
11	无法找到命令
12	解压缩文件错误
13	文件类型错误
14	根文件系统错误
15	文件检查错误
16	根分区跟备分区大小不一致

17	无法找到磁盘
18	卸载失败
19	安全加固执行错误
20	hook执行失败
22	depmod执行失败
23	mkinitrd执行失败
24	用户权限错误
25	日志分区检查错误
50	升级包不支持当前基线版本
51	无需回退
52	无需升级
53	/boot/与当前环境不匹配

hook脚本在升级时从ISO包中更新到本地，因此在做ISO包时将要执行的hook脚本放到custom/cfg_**/usr_file/opt/osupgrade/bin/hook这个目录下即可，也可将hook脚本直接放到/opt/osupgrade/bin/hook目录下，但注意脚本命名，不要与升级包中的脚本同名，避免被覆盖。