

Activate Upgrade

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2016, 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Description	1
2	Procedure	1
2.1	Activate Upgrade	1



Activate Upgrade



1 Description

This instruction describes how to perform the execution phase of a software upgrade.

A software upgrade has a limited time window during which some planned service impact is allowed. The execution phase of the software upgrade can have service impact and must be executed only within the upgrade time window and not during normal “traffic hours”.

The procedure in this instruction covers the following:

- How to activate (that is, apply) the upgrade package on the Managed Element (ME)
- How to confirm the upgrade

The procedure is illustrated by an example where a software version with product name `ERIC-COREMW_RUNTIME`, product number `CXP9020355_1`, and product revision `R7E01` is running in the system. A software upgrade package `ERIC_UP-CXP9020355_1-R7F01`, which is designed to upgrade this software version to product revision `R7F01`, has already been prepared and is going through the activation phase. This upgrade package has an activation fallback-timer, attribute `activationFallbackTimer` in the Managed Object (MO) *UpgradePackage* equal to 1200 seconds.

2 Procedure

2.1 Activate Upgrade

Prerequisites

- No documents are required.
- No tools are required.
- The following conditions must apply:
 - The ME has passed a health check routine.
 - The upgrade package is prepared.



- The fallback capability is supported by the ME, that is, attribute `timeoutFallbackCapability=SUPPORTED` in the *SwM* MO.
- An Ericsson Command-Line Interface (ECLI) session in Exec mode is in progress.

Steps

1. Navigate to the upgrade package, for example:

```
>dn ManagedElement=NODE06ST,SystemFunctions=1,SwM=1,UpgradePackage=ERIC_UP-CXP9020355_1-R7F01
```

2. Activate the first step of the upgrade package:

```
(UpgradePackage=ERIC_UP-CXP9020355_1-R7F01)>activate
```

The system returns output `true` for a successfully triggered activation or `false` otherwise.

If the upgrade package is configured for one-step activation, the first step is also the final step.

3. Verify the result for a successfully triggered activation:

```
(UpgradePackage=ERIC_UP-CXP9020355_1-R7F01)>show -v
```

In the following example output, `state=RUNNING` and `progressPercentage=10`. It shows that 10% of the activation is completed.

```
UpgradePackage=ERIC_UP-CXP9020355_1-R7F01
[...]
state=ACTIVATION_IN_PROGRESS <read-only>
[...]
reportProgress <read-only>
[...]
actionName="Activate" <read-only>
[...]
progressInfo="Activate UpgradePackage" <read-only>
progressPercentage=10 <read-only>
[...]
state=RUNNING <read-only>
[...]
```

4. Continue to check the progress of this activation step until it is completed:

```
(UpgradePackage=ERIC_UP-CXP9020355_1-R7F01)>show -v
```

The following example output shows the resulting state of an activation step, which is followed by at least one more activation step:



```
UpgradePackage=ERIC_UP-CXP9020355_1-R7F01
[...]
state=ACTIVATION_STEP_COMPLETED <read-only>
[...]
reportProgress <read-only>
[...]
progressInfo="Activate UpgradePackage" <read-only>
progressPercentage=30 <read-only>
result=SUCCESS <read-only>
resultInfo="Campaign execution suspended by breakpoint" <read-only>
state=FINISHED <read-only>
step=1 <read-only>
[...]
```

Note: State `ACTIVATION_STEP_COMPLETED` implicitly indicates that there is at least one more activation step to execute in this example.

The following example output shows the resulting state of a final activation step:

```
UpgradePackage=ERIC_UP-CXP9020355_1-R7F01
[...]
state=WAITING_FOR_COMMIT <read-only>
[...]
reportProgress <read-only>
[...]
progressInfo="Activate UpgradePackage" <read-only>
progressPercentage=100 <read-only>
result=SUCCESS <read-only>
resultInfo="Campaign execution successfully completed" <read-only>
state=FINISHED <read-only>
step=2 <read-only>
[...]
```

Note: State `WAITING_FOR_COMMIT` indicates that the final activation step has been executed successfully and that the upgrade can be confirmed.

5. Verify that the ME state or behavior is according to expectations at this breakpoint. This is specific for the software upgrade and outside the scope of this instruction.
6. Is the ME state or behavior according to expectations?

Yes: Continue with the next step.

No: Cancel the activation, refer to *Cancel Upgrade Operation*.

7. Select operation according to the value of parameter `state` in the output:
 - If `state=ACTIVATION_STEP_COMPLETED`, proceed with Step 2 to activate the next step.
 - If `state=WAITING_FOR_COMMIT`, continue with the next step.

8. Navigate to the *SwM* MO:

```
(UpgradePackage=ERIC_UP-CXP9020355_1-R7F01) >up
```

9. Verify how much time remains before fallback:



```
(SwM=1) >show timeRemainingBeforeFallback
```

The following example output shows that 1155 seconds remain before fallback:

```
1155
```



Do!

Carefully read the following information before proceeding.

Once the activation is effective, indicated by `state=WAITING_FOR_COMMIT`, the fallback countdown starts. Attribute `timeRemainingBeforeFallback` is automatically set to the value (in this example 1200 seconds) contained in attribute `fallbackTimer` in the `SwM MO`.

If the user does not confirm the operation before `timeRemainingBeforeFallback` reaches zero, a fallback is automatically triggered.

10. Navigate to the upgrade package:

```
(SwM=1) >UpgradePackage=ERIC_UP-CXP9020355_1-R7F01
```

11. Confirm the upgrade by using action `confirm` in the `Upgradepackage MO`:

```
(UpgradePackage=ERIC_UP-CXP9020355_1-R7F01) >confirm
```

The system returns `true` on success, `false` on failure.

Select operation according to the return value:

- If `confirm` returns `true` continue with the next step.
- If `confirm` returns `false` restore the backup, refer to `Restore Backup Operation`.

12. Check the result of action `confirm`:

```
(UpgradePackage=ERIC_UP-CXP9020355_1-R7F01) >show -v
```

The following is an example output when the `confirm` operation completed successfully:

```
[...]
state=COMMIT_COMPLETED
[...]
```




13. Navigate to the *SwInventory* MO:

```
(UpgradePackage=ERIC_UP-CXP9020355_1-R7F01) >up  
  
(SwM=1) >up  
  
(SystemFunctions=1) >SwInventory=1
```

14. Verify that the active software state is the expected one:

```
(SwInventory=1) >show -v
```

The following is an example output. The active software version corresponds to the activated software upgrade scope. A software version with product name `ERIC-COREMW_RUNTIME`, product number `CXP9020355_1`, and product revision `R7F01` is running in the system.

```
SwInventory=1  
  active <read-only>  
    "ManagedElement=NODE06ST, SystemFunctions=1, SwInventory=1, SwVersion=>  
ERIC-COREMW_RUNTIME-CXP9020355_1-R7F01"  
[...]  
  SwVersion=ERIC-COREMW_RUNTIME-CXP9020355_1-R7F01
```

A software version with product name `ERIC-COREMW_RUNTIME`, product number `CXP9020355_1`, and product revision `R7F01` is now confirmed and running in the system.