

Software Management

DESCRIPTION

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	Understanding Software Management	1
1.1	Key Software Management Concepts	1
1.2	Software Upgrade	1
1.3	Preparation Phase	2
1.4	Execution Phase	3
1.5	Fallback Operation	4
1.6	Software Upgrade Package Life Cycle	4
2	Basic Software Management Procedures	7
3	Software Management-Related Alarms	8





1 Understanding Software Management

1.1 Key Software Management Concepts

Software Management provides a management interface to upgrade and delete software installed on the Managed Element (ME).

The Software Management area is represented by a group of Managed Object Classes (MOCs) under *SwM* within the Managed Object Model (MOM). For general information about the MOM, MOCs, cardinality, and related concepts, refer to *Managed Object Model User Guide*.

1.2 Software Upgrade

A software upgrade is needed in the following situations:

- Adding new software domains or software items to the ME.
- Deleting or replacing software domains or software items on the ME.

For more information on software domains and software items, refer to *Software Inventory Management*.

A patch installation and a major software upgrade are both handled as software upgrades by Software Management.

A software upgrade is achieved through the installation of an Upgrade Package (UP). An UP is a collection of Load Modules and upgrade control information delivered by the Ericsson supply organization.

Depending on its design, a software UP can upgrade one or several software domains at a time.

Software management is able to perform upgrades using the upgrade execution methods: rolling, single-step, or balanced in-service. When applied to compatibly designed software domains and UPs, the upgrade execution method affects execution and ME services downtime as shown in Table 1.

Table 1 Software Upgrade Execution Methods

Execution Method	Effect
rolling	Upgrades one software service unit or node at a time. This method maximizes service capacity at the price of upgrade time; requires node wise compatibility.



Execution Method	Effect
single-step	Upgrades all nodes in one step. This method minimizes upgrade time at the price of total service outage.
balanced in-service	Upgrades nodes in the scalable portion of the cluster one group of nodes at a time. The number of nodes per group is calculated during UP execution according to available system resources and configured parameters. Non-scalable nodes are upgraded one node at a time, as per rolling execution method. This method aims to reach a balance between the upgrade time and the expected service capacity during the upgrade.

A software upgrade is divided into two main phases; a preparation phase and an execution phase. The preparation phase takes place during normal working hours. The execution phase takes place during “low traffic” hours.

1.3 Preparation Phase

The preparation phase has the following responsibilities:

- Verifies that the ME has passed a health check routine.
- Makes the UP visible in the MOM using metadata from the UP file.
- Downloads the necessary parts of the UP content (for example, required Load Modules) from an external server.
- Performs the relevant software UP extraction activities or integrity checks. The integrity check performs checksum checks or equivalent (oblivious hashing, check and guard system, active or passive tamper resistance) and is needed to secure that the correct upgrade package has been fetched.
- Verifies the ME ability to activate the UP. For example, it checks that the UP is consistent and matches the software version of the ME. This verification can also occur during the execution phase, depending on the design of the UP.
- Allows the user to configure execution phase behavior of the UP.

Perform pre-upgrade actions to bring the system to the state where it can consume the UP package successfully. For this purpose, the model compiler and the merge tool inside the UP package is used.



1.4 Execution Phase

The execution phase has the following responsibilities:

- Verifies that the ME has passed a health check routine.
- Verifies the ME ability to activate the UP (if it is designed in this way).
- If the UP is a CSP containing ESM configurations and the system has changed since the completion of the preparation phase, regenerates the upgrade campaign and the target configuration.
- Activates (that is, applies) the UP on the ME.
- Provides an observation window that allows the user to monitor the success of the upgrade.
- Triggers automatic or manual fallback in failure situations.
- Confirms the upgrade based on an explicit user request.
- Finalizes the target configuration.

The execution phase behaviors, configured during the prepare phase, are applied during activation of the UP. They are the following:

- Upgrade execution method: rolling, single-step, or balanced in-service.
- Activation: one-step or step-by-step.

An activation step is a breakpoint. It represents a part of the upgrade after which the ME functionality can be observed manually and require user interaction. An UP with multiple steps and therefore multiple breakpoints, enables the user to verify that each step has been correctly executed.

Step-by-step activation is applicable to UPs designed with multiple steps. It allows the user to check and interact with the ME after each step.

One-step activation is applicable to UPs designed with one or multiple steps. It is, however, not recommended to apply it to an UP designed for multiple steps, since it implies reduced user interaction and increases the upgrade failure probability.

Note: Single-step upgrade execution method always executes as per one-step activation, even if UP contains multiple steps and step-by-step activation is configured.

The first or only activation step triggers a system backup. It is recommended to keep this default behavior.

After successfully completing activation, the ME waits for a final confirmation, that is, confirmation from the user. The user confirms that observations of the ME functions have been completed satisfactorily. After a confirm, the user can no longer cancel an upgrade.



1.5 Fallback Operation

The upgrade activation procedure stops and the ME starts a fallback operation in the following cases:

- The ME encounters a critical problem that prohibits a successful activation.
- The user cancels the activation operation after an intermediary step or after the last step.
- The fallback timer expires during the activation. The fallback timer is the maximum number of seconds between the activate operation finishing and the next operation (activate or confirm) starting.

The ME raises the alarm `A Fallback Operation will soon be started` to indicate these conditions.

The fallback operation triggers a backup restore, which restores the ME to the state it had before the activation procedure started. It is recommended to keep this default behavior. The fallback operation is not supposed to fail under any circumstances.

1.6 Software Upgrade Package Life Cycle

During the preparation and execution phases, a software UP goes through different life cycle states.

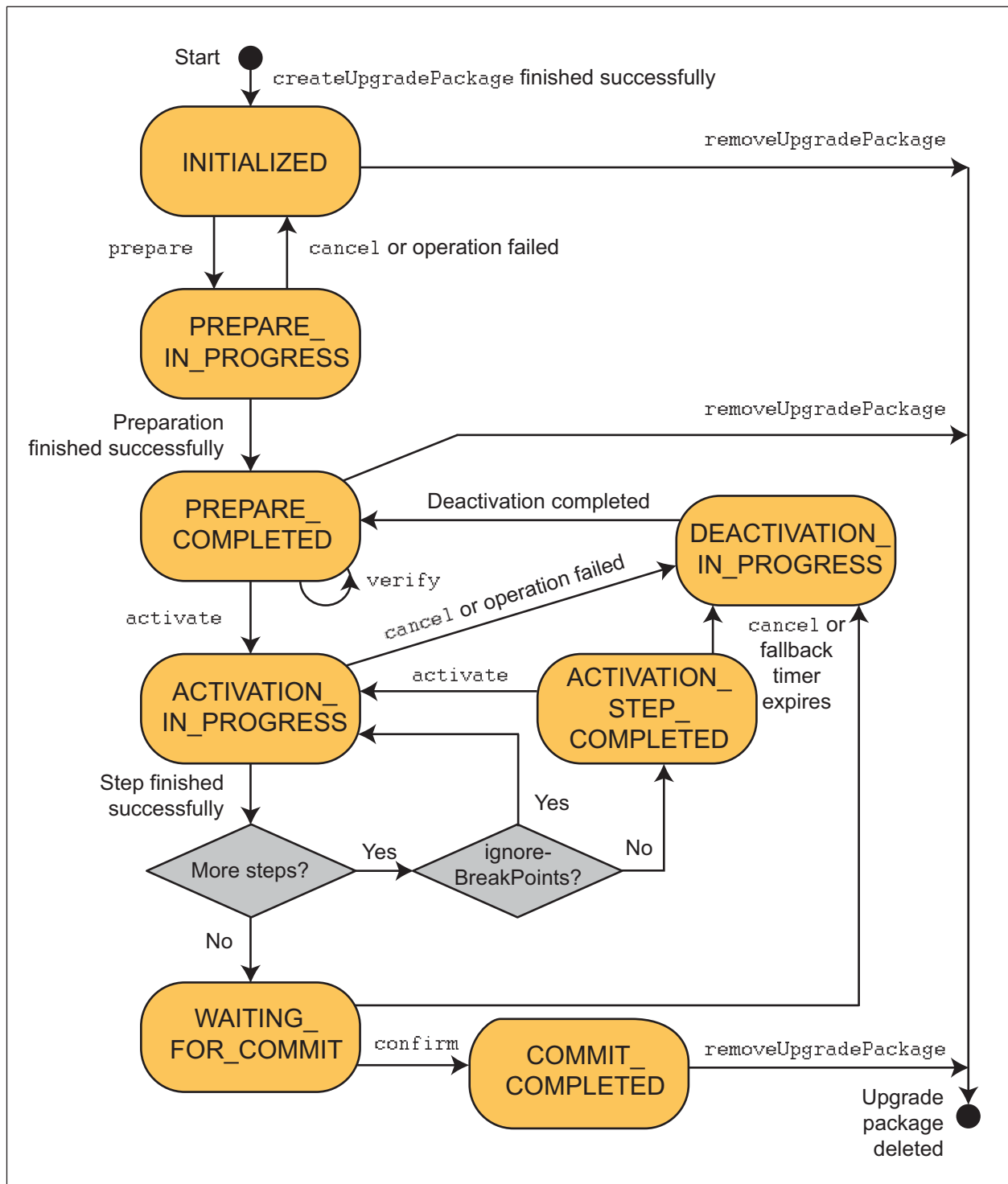


Figure 1 Life Cycle of an Upgrade Package

Software Upgrade

— Prepare UP



The user can initiate a UP preparation in the following two ways:

- Choose to specify over the Northbound Interface (NBI) from which Uniform Resource Identifier (URI) to retrieve the UP file.
- Download the UP file on the ME using the SSH File Transfer Protocol (SFTP).

The user can configure the execution phase of the UP.

The procedure in [Prepare Upgrade Package](#) provides further details on how to perform this operation.

— Activate UP

The user can activate the execution phase of the UP to apply and confirm the software upgrade. The procedure in [Activate Upgrade](#) provides further details on how to perform this operation.

— Cancel an upgrade operation

The user can perform different types of cancel operations:

- Cancel an ongoing prepare action.
- Cancel an ongoing activation step.
- Cancel a completed intermediary activation step during a step-by-step activation.
- Cancel the final and completed activation step (before action confirm).

The procedure in [Cancel Upgrade Operation](#) provides further details on how to perform this operation.

An ongoing operation is usually canceled when it has been started by mistake, at the wrong time, or takes too long time.

A completed activation step is usually canceled when the user observation and assessment of the ME health check status is not according to expectations.

— Delete a UP

Once a UP has been applied and is no longer needed on the ME, the user can delete the corresponding Managed Object (MO). The procedure in [Delete Upgrade Package](#) provides further details on how to perform this operation.

If the upgrade file initially was retrieved from a URI, this operation deletes it together with the MO. If the UP file initially was downloaded with SFTP, this operation does not delete it from the ME.

Software Removal

— Delete an inactive software version



The user can delete an inactive software version from the ME. This operation can be used to save disk space when an inactive software version, which is no longer needed, has not been deleted automatically by the latest software upgrade. The procedure in [Delete Inactive Software Version](#) provides further details on how to perform this operation.

Any attempt to delete an active software version fails.

To gain disk storage space, the user can delete an inactive software version from the ME. An inactive software version is deleted with care. That is, software items shared with other software versions are not deleted. An upgrade can also automatically delete an obsolete software version from the ME.

The active software version cannot be deleted.

2 Basic Software Management Procedures

Software Management is accessed using NETCONF or the Ericsson Command-Line Interface (ECLI) to manipulate the Management Information Base (MIB).

The following operations can be performed by the user and are described in Operating Instructions using the ECLI:

Software Upgrade

- Prepare Upgrade Package
- Activate Upgrade
- Cancel Upgrade Operation
- Delete Upgrade Package

Software Removal

- Delete Inactive Software Version



3 Software Management-Related Alarms

Table 2 Software Management-Related Alarms

Alarm	Description
A Fallback Operation will soon be started	Issued during a software upgrade process when waiting on action <code>confirm</code> or <code>activate</code> to prevent the upgrade from being automatically canceled.