

Software Management

DESCRIPTION

Copyright

© Ericsson AB 2017, 2018. 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 | Introduction | 1 |
| 2 | Functions and Concepts | 3 |
| 2.1 | Types of Operation | 7 |
| 3 | Managed Object Model | 9 |
| 4 | Configuration Management | 11 |
| 5 | Fault Management | 13 |
| | Reference List | 15 |





1 Introduction

This document provides an overview of the management model and concepts associated with the Software Management managed area.

A managed area is represented by a group of Managed Object Classes (MOCs) within the Managed Object Model (MOM).





2 Functions and Concepts

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

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 rolling upgrades, that is, software upgrades that are not causing any downtime to the ME services. Rolling upgrades are applicable only for software domains and UPs designed for this purpose.

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.

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.

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).
- Activates (that is, applies) the UP on the ME.
- Provides an observation window that allows the user to monitor the success of the upgrade.
- Commit the upgrade based on an explicit user request.

The execution phase can be achieved using the one-step activation, depending on the design of the UP.

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

A 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 enable the user to verify that each step has been correctly executed.

After successfully completing a one-step activation, the ME waits for a final confirmation (that is, commit) from the user. With a commit, the user confirms that observations of the ME functions have been completed satisfactorily.

Fallback Operation

Graceful fallback is not supported. When upgrade fails, refer to the section *Rollback Activities* in IPWorks Upgrade Instruction , see Reference [1].

Software Upgrade Package Life Cycle

During the preparation and execution phases, a software UP goes through different life cycle states. An overview is provided in Figure 1.



Software Removal

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.

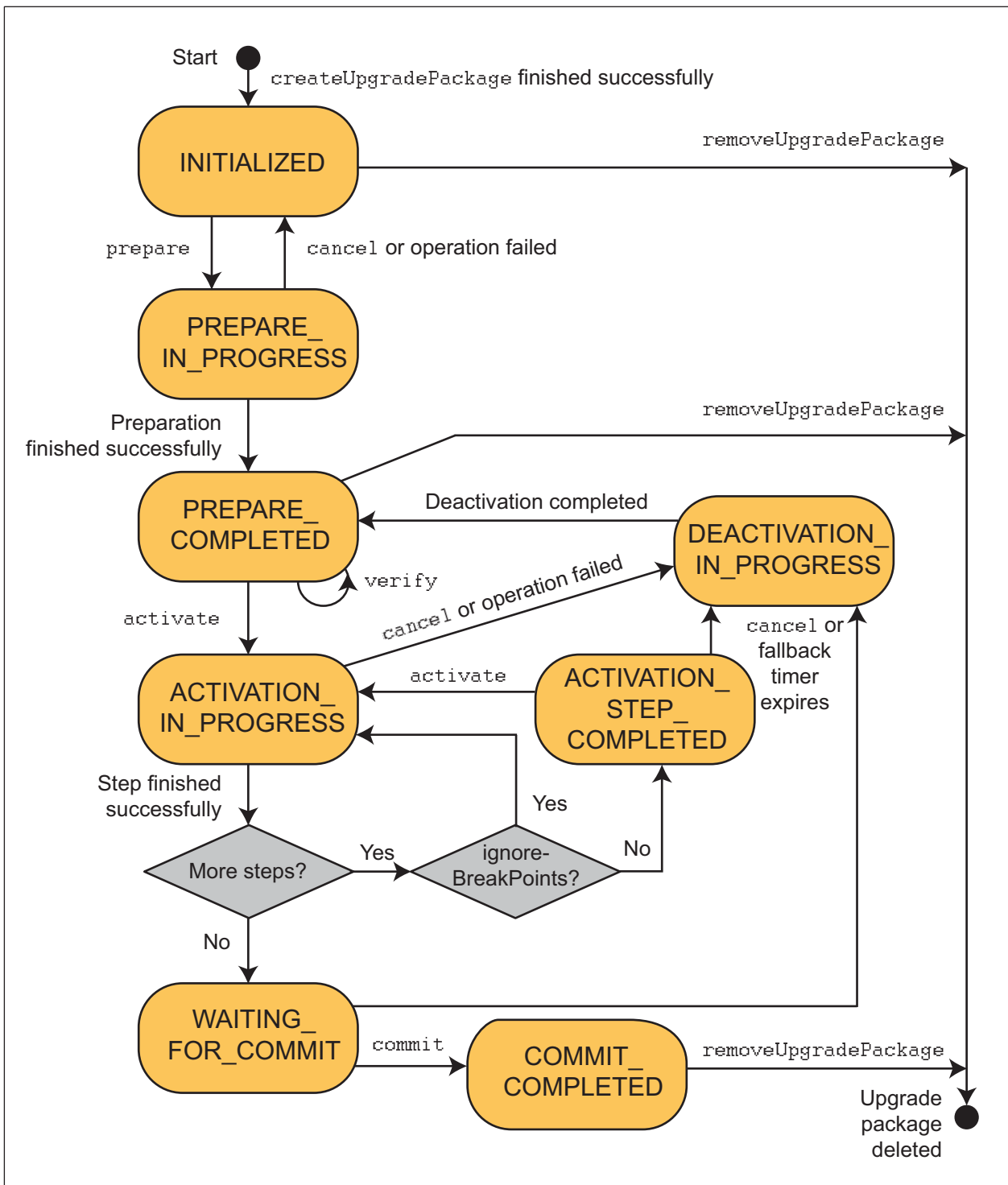


Figure 1 Life Cycle of an Upgrade Package



2.1 Types of Operation

Software Management supports the following operations:

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.

The procedure in *Prepare Upgrade Package* provides further details on how to perform this operation.

- Upgrade using one-step activation

This operation is applicable to UPs designed for one-step activations. However, it is not recommended to apply it to a UP designed for multiple steps, since it implies reduced user interaction and increases the upgrade failure probability. The procedure in *Upgrade Using One-Step Activation* provides further details on how to perform this operation.

- Cancel operation is not supported. When upgrade fails, refer to the section *Rollback Activities* in IPWorks Upgrade Instruction , see Reference [1].

- 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.



3 Managed Object Model

The Software Management managed area is represented in the *Managed Object Model (MOM)* as follows:

```
ManagedElement
+-SystemFunctions
+-SwM
+-SwVersionMain
+-UpgradePackage
```

For general information about the MOM, MOCs, MOs, cardinality, and related concepts, refer to *Managed Object Model User Guide*.

The Software Management MOCs are described in Table 1.

Table 1 Software Management Managed Object Class Descriptions

| Managed Object Class | Description |
|-----------------------|---|
| <i>SwM</i> | The root of the Software Management model. Supports the creation and deletion of <i>UpgradePackage</i> MOs, the deletion of inactive software versions. Provides control of some software management capabilities such as automatic backup and automatic restore. Indicates the active software versions. |
| <i>SwVersionMain</i> | Indicates which software versions are present on the ME. |
| <i>UpgradePackage</i> | Handles the preparation phase and the execution phase for a UP except the <i>UpgradePackage</i> MO creation. |





4 Configuration Management

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*
- *Upgrade Using One-Step Activation*
- *Delete Upgrade Package*

Software Removal

- *Delete Inactive Software Version*





5 Fault Management

The Software Management alarm is described in Table 2.

Table 2 Software Management Alarm

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





Reference List

Ericsson Documents

- [1] *IPWorks Upgrade Instruction*, 1/153 72-AVA 901 33/2 Uen