3GPP TS 32.332 V16.0.0 (2020-07)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Telecommunication management;

Notification Log (NL) Integration Reference Point (IRP);

Information Service (IS)

(Release 16)

* *

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.   
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.   
This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

UMTS, Management

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword [6](#__RefHeading___Toc517803127)

Introduction [6](#__RefHeading___Toc517803128)

1 Scope [7](#__RefHeading___Toc517803129)

2 References [7](#__RefHeading___Toc517803130)

3 Definitions and abbreviations [7](#__RefHeading___Toc517803131)

3.1 Definitions [7](#__RefHeading___Toc517803132)

3.2 Abbreviations [8](#__RefHeading___Toc517803133)

4 System overview [8](#__RefHeading___Toc517803134)

4.2 Compliance rules [8](#__RefHeading___Toc517803135)

5 Information Object Classes (IOCs) [9](#__RefHeading___Toc517803136)

5.1 Information entities imported and local labels [9](#__RefHeading___Toc517803137)

5.2 Class diagram [10](#__RefHeading___Toc517803138)

5.2.1 Attributes and relationships [10](#__RefHeading___Toc517803139)

5.2.2 Inheritance [11](#__RefHeading___Toc517803140)

5.3 Information Object Class (IOC) definitions [12](#__RefHeading___Toc517803141)

5.3.1 NLIRP [12](#__RefHeading___Toc517803142)

5.3.1.1 Definition [12](#__RefHeading___Toc517803143)

5.3.1.2 Attributes [12](#__RefHeading___Toc517803144)

5.3.2 Log [12](#__RefHeading___Toc517803145)

5.3.2.1 Definition [12](#__RefHeading___Toc517803146)

5.3.2.2 Attributes [12](#__RefHeading___Toc517803147)

5.3.2.3 State diagram [13](#__RefHeading___Toc517803148)

5.3.3 LogRecord [13](#__RefHeading___Toc517803149)

5.3.3.1 Definition [13](#__RefHeading___Toc517803150)

5.4 Information relationship definitions [14](#__RefHeading___Toc517803151)

5.4.1 Relation-nLIRP-log (M) [14](#__RefHeading___Toc517803152)

5.4.1.1 Definition [14](#__RefHeading___Toc517803153)

5.4.1.2 Role [14](#__RefHeading___Toc517803154)

5.4.1.3 Constraint [14](#__RefHeading___Toc517803155)

5.4.2 Relation-log-logRecord (M) [14](#__RefHeading___Toc517803156)

5.4.2.1 Definition [14](#__RefHeading___Toc517803157)

5.4.2.2 Role [14](#__RefHeading___Toc517803158)

5.4.2.3 Constraint [14](#__RefHeading___Toc517803159)

5.4.3 Relation-logRecord-notificationIRPNotification (M) [15](#__RefHeading___Toc517803160)

5.4.3.1 Definition [15](#__RefHeading___Toc517803161)

5.4.3.2 Role [15](#__RefHeading___Toc517803162)

5.4.3.3 Constraint [15](#__RefHeading___Toc517803163)

5.5 Information attribute definition [16](#__RefHeading___Toc517803164)

5.5.1 Definition and legal values [16](#__RefHeading___Toc517803165)

5.5.2 Constraints [17](#__RefHeading___Toc517803166)

6 Interface definition [17](#__RefHeading___Toc517803167)

6.1 Class diagram [17](#__RefHeading___Toc517803168)

6.2 Generic rules [18](#__RefHeading___Toc517803169)

6.3 NLIRPOperations\_1 Interface (M) [18](#__RefHeading___Toc517803170)

6.3.1 Operation subscribeLog (M) [18](#__RefHeading___Toc517803171)

6.3.1.1 Definition [18](#__RefHeading___Toc517803172)

6.3.1.2 Input parameters [18](#__RefHeading___Toc517803173)

6.3.1.3 Output parameters [19](#__RefHeading___Toc517803174)

6.3.1.4 Pre-condition [19](#__RefHeading___Toc517803175)

6.3.1.5 Post-condition [19](#__RefHeading___Toc517803176)

6.3.1.6 Exceptions [19](#__RefHeading___Toc517803177)

6.3.2 Operation unsubscribeLog (M) [20](#__RefHeading___Toc517803178)

6.3.2.1 Definition [20](#__RefHeading___Toc517803179)

6.3.2.2 Input parameters [20](#__RefHeading___Toc517803180)

6.3.2.3 Output parameters [20](#__RefHeading___Toc517803181)

6.3.2.4 Pre-condition [20](#__RefHeading___Toc517803182)

6.3.2.5 Post-condition [20](#__RefHeading___Toc517803183)

6.3.2.6 Exceptions [20](#__RefHeading___Toc517803184)

6.3.3 Operation exportLogRecords (O) [21](#__RefHeading___Toc517803185)

6.3.3.1 Definition [21](#__RefHeading___Toc517803186)

6.3.3.2 Input parameters [21](#__RefHeading___Toc517803187)

6.3.3.3 Output parameters [21](#__RefHeading___Toc517803188)

6.3.3.4 Pre-condition [21](#__RefHeading___Toc517803189)

6.3.3.5 Post-condition [21](#__RefHeading___Toc517803190)

6.3.3.6 Exceptions [21](#__RefHeading___Toc517803191)

6.3.4 Operation getLogRecords (O) [22](#__RefHeading___Toc517803192)

6.3.4.1 Definition [22](#__RefHeading___Toc517803193)

6.3.4.2 Input parameters [22](#__RefHeading___Toc517803194)

6.3.4.3 Output parameters [22](#__RefHeading___Toc517803195)

6.3.4.4 Pre-condition [22](#__RefHeading___Toc517803196)

6.3.4.5 Post-condition [22](#__RefHeading___Toc517803197)

6.3.4.6 Exceptions [22](#__RefHeading___Toc517803198)

6.4 NLIRPOperations\_2 Interface (O) [23](#__RefHeading___Toc517803199)

6.4.1 Operation getLogSubscriptionIds (M) [23](#__RefHeading___Toc517803200)

6.4.1.1 Definition [23](#__RefHeading___Toc517803201)

6.4.1.2 Input parameters [23](#__RefHeading___Toc517803202)

6.4.1.3 Output parameters [23](#__RefHeading___Toc517803203)

6.4.1.4 Pre-condition [23](#__RefHeading___Toc517803204)

6.4.1.5 Post-condition [23](#__RefHeading___Toc517803205)

6.4.1.6 Exceptions [23](#__RefHeading___Toc517803206)

6.4.2 Operation getLogSubscriptionStatus (M) [24](#__RefHeading___Toc517803207)

6.4.2.1 Definition [24](#__RefHeading___Toc517803208)

6.4.2.2 Input parameters [24](#__RefHeading___Toc517803209)

6.4.2.3 Output parameters [24](#__RefHeading___Toc517803210)

6.4.2.4 Pre-condition [24](#__RefHeading___Toc517803211)

6.4.2.5 Post-condition [24](#__RefHeading___Toc517803212)

6.4.2.6 Exceptions [24](#__RefHeading___Toc517803213)

6.5 NLIRPNotifications\_1 Interface (M) [25](#__RefHeading___Toc517803214)

6.5.1 Notification notifyLogSubscribed (M) [25](#__RefHeading___Toc517803215)

6.5.1.1 Definition [25](#__RefHeading___Toc517803216)

6.5.1.2 Input Parameters [25](#__RefHeading___Toc517803217)

6.5.1.3 Triggering Event [25](#__RefHeading___Toc517803218)

6.5.1.3.1 From-state [25](#__RefHeading___Toc517803219)

6.5.1.3.2 To-state [25](#__RefHeading___Toc517803220)

6.5.2 Notification notifyLogUnsubscribed (M) [26](#__RefHeading___Toc517803221)

6.5.2.1 Definition [26](#__RefHeading___Toc517803222)

6.5.2.2 Input Parameters [26](#__RefHeading___Toc517803223)

6.5.2.3 Triggering Event [26](#__RefHeading___Toc517803224)

6.5.2.3.1 From-state [26](#__RefHeading___Toc517803225)

6.5.2.3.2 To-state [26](#__RefHeading___Toc517803226)

6.6 NLIRPNotifications\_2 Interface (O) [27](#__RefHeading___Toc517803227)

6.6.1 Notification notifyOccupancyLevelCrossed (M) [27](#__RefHeading___Toc517803228)

6.6.1.1 Definition [27](#__RefHeading___Toc517803229)

6.6.1.2 Input Parameters [27](#__RefHeading___Toc517803230)

6.6.1.3 Triggering Event [27](#__RefHeading___Toc517803231)

6.6.1.3.1 From-state [27](#__RefHeading___Toc517803232)

6.6.1.3.2 To-state [27](#__RefHeading___Toc517803233)

6.6.2 Notification notifyLoggingResumed (O) [28](#__RefHeading___Toc517803234)

6.6.2.1 Definition [28](#__RefHeading___Toc517803235)

6.6.2.2 Input Parameters [28](#__RefHeading___Toc517803236)

6.6.2.3 Triggering Event [28](#__RefHeading___Toc517803237)

6.6.2.3.1 From-state [28](#__RefHeading___Toc517803238)

6.6.2.3.2 To-state [28](#__RefHeading___Toc517803239)

Annex A (informative): Change history [29](#__RefHeading___Toc517803240)

# Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

# Introduction

The present document is part of a TS-family covering the 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; as identified below:

TS 32.331 "Notification Log (NL) Integration Reference Point (IRP): Requirements"

**TS 32.332 "Notification Log (NL) Integration Reference Point (IRP): Information Service (IS)"**

TS 32.336 "Notification Log (NL) Integration Reference Point (IRP); Solution Set (SS) definitions"

The present document describes the requirements and information model necessary for Telecommunication Management (TM). The TM principles and TM architecture are specified in 3GPP TS 32.101 [1] and 3GPP TS 32.102 [2].

A communications system is composed of a multitude of Network Elements (NE) of various types and, typically, different vendors, which inter-operate in a co-ordinated manner in order to satisfy the network users' communication requirements.

The occurrence of faults in an NE may cause deterioration or loss of this NE's function. Fault Management is the functional area, which allows the operator to detect the occurrence of faults in the network in real-time. Configuration Management and Performance Management are two more functional areas, which require the operator to be alerted to certain conditions in the network.

A standard general-purpose mechanism for the management of logs containing selected notifications from the network is required to provide an ability to perform historical analysis on faults and conditions, which occurred in the network.   
The TS 32.33x-series, constituting the Notification log IRP, sets forth such a mechanism - and the present document contains the IS definition.

# 1 Scope

The present document specifies the Information Service for the Notification Log Integration Reference Point (NLIRP) as it applies to Itf-N.

This IRP IS defines the semantics of operations (and their parameters) visible across the Itf-N in a protocol and technology neutral way. It does not define the syntax or encoding of the operations and their parameters.

# 2 References

The following documents contain provisions, which through reference in this text constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2] 3GPP TS 32.102: "Telecommunication management; Architecture".

[3] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information Service (IS)".

[4] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[5] 3GPP TS 32.111-2: "Telecommunication management; Fault Management; Part 2: Alarm Integration Reference Point (IRP): Information Service (IS)".

[6] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information Service (IS)".

[7] 3GPP TS 32.331: "Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Requirements".

[8] 3GPP TS 32.342 "Telecommunication management; File Transfer (FT) Integration Reference Point (IRP): Information Service (IS)".

[9] 3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.101 [1], 3GPP TS 32.102 [2], 3GPP TS 32.111-2 [5] and 3GPP TS 32.331 [7] apply.

**notification category:** it refers to the set of notifications of one 3GPP IRP Information Service specification  
A Notification Category is identified by the name of the IRP specification and the IRP specification version number.

## 3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

DN Distinguished Name

EM Element Manager

FT File Transfer

IOC Information Object Class

IRP Integration Reference Point

M Mandatory

NE Network Element

NL Notification Log

NLIRP Notification Log Integration Reference Point

NM Network Manager

O Optional

UML Unified Modelling Language

# 4 System overview

The general definition of the System Context for the present IRP is found in 3GPP TS 32.150 [9] clause 4.7.

In addition, the set of related IRP(s) relevant to the present IRP is shown in the two diagrams below



Figure 4.1: System Context A



Figure 4.2: System Context B

## 4.2 Compliance rules

For general definitions of compliance rules related to qualifiers (Mandatory/Optional/Conditional) for *operations*, *notifications* *and* *parameters* (of operations and notifications) please refer to 3GPP TS 32.102 [2].

# 5 Information Object Classes (IOCs)

## 5.1 Information entities imported and local labels

|  |  |
| --- | --- |
| Label reference | Local label |
| 3GPP TS 32.622 [4], information object class, Top | Top |
| 3GPP TS 32.312 [6], information object class, ManagedGenericIRP | ManagedGenericIRP |
| 3GPP TS 32.342 [8], information object class, FileTransferIRP | FileTransferIRP |
| 3GPP TS 32.302 [3], information object class, NotificationIRP | NotificationIRP |
| 3GPP TS 32.302 [3], information object class, NotificationIRPNotification | NotificationIRPNotification |

## 5.2 Class diagram

### 5.2.1 Attributes and relationships

This clause introduces the set of IOCs that encapsulate information within the IRPAgent. The intent is to identify the information required for NLIRP operations and notification. This clause provides the overview of all support object classes in UML. Subsequent clauses provide more detailed specification of various aspects of these support object classes.

NotificationIRPNotification

notificationId

eventTime

systemDN

notificationType

(from TS32.302)

LogRecord

logRecordId

logRecordContent

<<InformationObjectClass>>

1

1

+notification

1

+logRecord

1

*relation-logRecord-notification*

Log

logSubscriptionId

loggingEndTime

maxSize

currentSize

creationTime

logState

logRecordCount

notificationCategories

filter

logFullAction

occupancyLevels

logManagerToken

<<InformationObjectClass>>

0..\*

1

+logRecord

0..\*

+log

1

*relation-log-logRecord*

NLIRP

maxLogs

<<InformationObjectClass>>

0..\*

0..\*

<<names>>

Figure 5.1: Information Object Class (IOC) UML diagram

To change: change logid, add logrecordcontent, remove logrecordsource.

### 5.2.2 Inheritance

ManagedGenericIRP

(from TS32.312)

Top

objectClass

objectInstance

(from TS32.622)

<<InformationObjectClass>>

NLIRP

maxLogs

<<InformationObjectClass>>

LogRecord

logRecordId

logRecordContent

<<InformationObjectClass>>

Log

logSubscriptionId

loggingEndTime

maxSize

currentSize

creationTime

logState

logRecordCount

notificationCategories

filter

logFullAction

occupancyLevels

logManagerToken

<<InformationObjectClass>>

Figure 5.2: Information Object Class (IOC) inheritance UML diagram

## 5.3 Information Object Class (IOC) definitions

### 5.3.1 NLIRP

#### 5.3.1.1 Definition

LogIRP is the representation of the notification log management capabilities specified by the present document. This IOC inherits from ManagedGenericIRP IOC specified in TS 32.312 [6].

#### 5.3.1.2 Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Support Qualifier | Read Qualifier | Write Qualifier |
| maxLogs | O | M | - |

### 5.3.2 Log

#### 5.3.2.1 Definition

The Log IOC is the representation of a Notification Log.

#### 5.3.2.2 Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Support Qualifier | Read Qualifier | Write Qualifier |
| logSubscriptionId | M | M | - |
| loggingEndTime | O | M | - |
| logManagerToken | O | - | - |
| maxSize | O | M | - |
| currentSize | O | M | - |
| creationTime | O | M | - |
| logState | M | M | - |
| logRecordCount | O | M | - |
| notificationCategories | O | M | - |
| filter | O | M | - |
| logFullAction | M | M | - |
| occupancyLevels | M | M | - |

#### 5.3.2.3 State diagram



Figure 5.3: State diagram for Notification Log

The disposition of a log that has been stopped, that is, whether the log remains visible across the Itf-N, is left as vendor specific functionality. The time of the deletion of logs is vendor specific.

### 5.3.3 LogRecord

#### 5.3.3.1 Definition

The LogRecord IOC is the representation of an individual Notification Log Record.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute name | Support Qualifier | Read Qualifier | Write Qualifier |
| logRecordId | M | M | - |
| logRecordContent | O | M | - |

## 5.4 Information relationship definitions

### 5.4.1 Relation-nLIRP-log (M)

#### 5.4.1.1 Definition

This represents the relationship between NLIRP and the Log.

#### 5.4.1.2 Role

|  |  |
| --- | --- |
| Name | Definition |
| nLIRP | It represents the NLIRP. |
| log | It represents the Log. |

#### 5.4.1.3 Constraint

|  |  |
| --- | --- |
| Name | Definition |
| uniqueLogSubscriptionId | The log subscription id must be unique amongst all logs managed by a given NLIRP instance. |
| uniqueLogManagerToken | The log manager token must be unique amongst all managers and logs utilizing logging services from a given NLIRP instance. |

### 5.4.2 Relation-log-logRecord (M)

#### 5.4.2.1 Definition

This represents the relationship between Log and the LogRecord.

#### 5.4.2.2 Role

|  |  |
| --- | --- |
| Name | Definition |
| log | It represents the Log. |
| logRecord | It represents the LogRecord. |

#### 5.4.2.3 Constraint

|  |  |
| --- | --- |
| Name | Definition |
| uniqueLogRecordId | The log record id must be unique amongst all logs records within a given log. |

### 5.4.3 Relation-logRecord-notificationIRPNotification (M)

#### 5.4.3.1 Definition

This represents the relationship between LogRecord and the notification header represented by NotificationIRPNotification.

#### 5.4.3.2 Role

|  |  |
| --- | --- |
| Name | Definition |
| logRecord | It represents the LogRecord. |
| notification | It represents the NotificationIRPNotification. |

#### 5.4.3.3 Constraint

|  |  |
| --- | --- |
| Name | Definition |
| logRecordIdRelatesNotificationId | Within a given log, there is a one-to-one relationship between Notification Id and Log Id (as each notification can only be recorded once). |

## 5.5 Information attribute definition

### 5.5.1 Definition and legal values

| Attribute Name | Definition | Legal Values |
| --- | --- | --- |
| creationTime | The time when the log is created. | YYYYMMDDhhmmss |
| currentSize | This attribute provides the number of bytes currently utilized by a given log. When taken in conjunction with maxSize, the amount of space remaining in the log can be determined. | Either:  - zero  - a positive whole number |
| filter | It specifies a filter constraint that IRPAgent shall use to filter notification of the category specified in notificationCategories parameter.  Support of time based filter is mandatory. Support of other filter constraints is optional. | Filter constraint grammar is SS dependent |
| logFullAction | Indicate the action that will be taken by this instance of NLIRP when the  Log.maxSize has been reached. | An ENUM that can have one of the following values:  - wrap: The oldest LogRecord(s) in the Log, based on the log time, will be deleted to free resources for the logging of new LogRecord(s).  - halt: No more LogRecord(s) will be logged and all incoming events are discarded. LogRecord/s already in the Log will be retained. |
| loggingEndTime | Defines the date and time when the log stops logging. | If log is created by IRPManager:   * value provided by the IRP Manager - the IRP Agent may reject the value provided by the IRP Manager in case this value is excessive * in case no value is provided by the IRP Manager then this value is set by the IRP Agent.   If log is created by IRPAgent:   * value indicates date and time or * value is expressed as “indefinite”. |
| logManagerToken | This attribute contains the token of a manager utilizing logging services from a given NLIRP instance, assigned by the IRPAgent. | The value of this attribute must be unique amongst all managers and logs utilizing logging services from the given NLIRP instance. |
| logRecordContent | The notification information of a log record, excluding the notification header information. | The content format varies depending on the type of the notification. |
| logRecordCount | The number of log records currently logged within a given log. | positive whole number, including zero |
| logRecordId | This attribute contains the id of a log record within a given log, assigned by the IRPAgent. | The value of this attribute must be unique amongst all log record contained by a given log. |
| logState | Provides an indication of the current state of a specific log | An ENUM that can have one of the following values:   * logging * logFull * stopped |
| logSubscriptionId | This attribute contains the id of a log subscription within a given NLIRP instance, assigned by the IRPAgent. | The value of this attribute must be unique amongst all log subscriptions managed by a given NLIRP instance. |
| maxLogs | Defines the maximum number of logs that can be supported by a given Notification Log IRP, assigned by the IRPAgent. | Non-zero, positive whole number |
| maxSize | This attribute defines the maximum number of bytes that may be utilized by a given log, assigned by the IRPAgent. | Either:   * Non-zero, positive whole number * Zero indicates no limit on the number of records is set by the IRPAgent (log full handling will provide limitation information) |
| notificationCategories | Specifies the notification categories that can be recorded within a given log - SET OF (name of IRP, version of IRP) | It identifies one or more Notification Category (see also Definition in clause 3.1) |
| occupancyLevels | Case Log.logFullAction == ‘halt’ and Log.maxSize is non-zero:   * this attribute contains a list of 3 values fixed (determined by the IRPAgent and never changed) of percentage, of Log.maxLog * the NLIRP will generate notifyOccupancyLevelCrossed to alert IRPManagers about the Log capacity currently used.   Case Log.logFullAction == ‘wrap’:   * this attribute contains 1 value fixed (determined by the IRPAgent and never changed) of percentage of Log.maxLog   the NLIRP will generate notifyOccupancyLevelCrossed to alert IRPManagers that the specified percentage of the capacity has been written to the wrapping Log. | Case of ‘halt’: a set of 3 values.  Case of ‘wrap’: 1 value. |

### 5.5.2 Constraints

|  |  |
| --- | --- |
| Name | Definition |
| inv\_lastModifiedTime | Time indicated shall be greater than or equal to creationTime |
| inv\_logRecordCount | Number indicated shall be less than or equal to maxSize |

# 6 Interface definition

## 6.1 Class diagram

NLIRPOperations\_1

subscribeLog()

unsubscribeLog()

<<optional>> exportLogRecords()

<<optional>> getLogRecords()

<<Interface>>

FileTransferIRP

(from TS32.342)

NLIRPNotifications\_1

notifyLogSubscribed()

notifyLogUnsubscribed()

<<Notification>>

NLIRPOperations\_2

getLogSubscriptionIds()

getLogSubscriptionStatus()

<<Interface>>

NLIRPNotifications\_2

notifyOccupancyLevelCrossed()

<<optional>> notifiyLoggingResumed()

<<Notification>>

NLIRP

maxLogs

<<InformationObjectClass>>

<<use>>

<<may realize>>

<<agent-internal-usage>>

<<may use>>

NotificationIRP

(from TS32.302)

<<InformationObjectClass>>

<<agent-internal-usage>>

<<agent-internal-usage>>

NLIRPOperations\_1 shall support either

operation exportLogRecords or operation

getLogRecords as mandatory.

Figure 6.1: Class diagram

NLIRPOperations\_1 shall support either operation exportLogRecords or operation getLogRecords as mandatory.

## 6.2 Generic rules

**Rule 1:** Each operation with at least one input parameter supports a pre-condition valid\_input\_parameter which indicates that all input parameters shall be valid with regards to their information type. Additionally, each such operation supports an exception operation\_failed\_invalid\_input\_parameter which is raised when pre-condition valid\_input\_parameter is false. The exception has the same entry and exit state.

**Rule 2:** Each operation with at least one optional input parameter supports a set of pre-conditions supported\_optional\_input\_parameter\_xxx where "xxx" is the name of the optional input parameter and the pre-condition indicates that the operation supports the named optional input parameter. Additionally, each such operation supports an exception operation\_failed\_unsupported\_optional\_input\_parameter\_xxx which is raised when (a) the pre-condition supported\_optional\_input\_parameter\_xxx is false and (b) the named optional input parameter is carrying information. The exception has the same entry and exit state.

**Rule 3:** Each operation shall support a generic exception operation\_failed\_internal\_problem which is raised when an internal problem occurs and that the operation cannot be completed. The exception has the same entry and exit state.

## 6.3 NLIRPOperations\_1 Interface (M)

### 6.3.1 Operation subscribeLog (M)

#### 6.3.1.1 Definition

Using this operation, an IRPManager is initiating the logging of notifications. Resulting from this operation an IRPAgent shall start logging of notifications and, if necessary, also create an associated log.

#### 6.3.1.2 Input parameters

| Parameter Name | Qualifier | Information type | Comment |
| --- | --- | --- | --- |
| logSubscriptionId | M | Log.logSubscriptionId | See clause 5.5.1  If empty, then IRPAgent shall create a new log and return the logSubscriptionId as output parameter. However not all IRPAgents will provide this capability, i.e. IRPManager must provide an existing logSubscriptionId. |
| loggingEndTime | O | Log.loggingEndTime | See clause 5.5.1 |
| notificationCategories | O | Log.notificationCategories | See clause 5.5.1; if notificationCategories is absent than all notifications are logged |
| filter | O | Log.ntfFilterfilter  Filter constraint grammar is SS dependent | See clause 5.5.1; if this parameter is absent, then no filter constraint shall be applied. |

#### 6.3.1.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| logSubscriptionId | M | Log.logSubscriptionId | See clause 5.5.1  This parameter contains either   * the logSubscriptionId of the log created resulting from this operation or * the value of the input parameter logSubscriptionId. |
| logManagerToken | O | Log.logManagerToken | See clause 5.5.1  See also comment 6.3.2.2.  Note: Security Management IRP may provide capabilities that make this parameter redundant. |
| loggingEndTime | O | Log.loggingEndTime | See clause 5.5.1  If supported by the Log IOC (5.3.2) and not already provided by the IRP Manager then the output of this operation must support this parameter. |
| status | M | ENUM (OperationSucceeded, OperationFailed) | If loggingEndTime is valid and notificationCategoriesis valid or absent, status = OperationSucceeded.  If operation\_failed is true, status = OperationFailed.  If loggingEndTime excessive, status = OperationFailed. |

#### 6.3.1.4 Pre-condition

logsNotMaxed

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logsNotMaxed | The number of logs is less than the maximum number of logs allowed. |

#### 6.3.1.5 Post-condition

logStarted

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logStarted | A log is started with the specified characteristics (lifetime and notificationCategories). |

#### 6.3.1.6 Exceptions

| Exception Name | Definition |
| --- | --- |
| logSubscriptionId\_required | **Condition:** Pre-condition is true AND post-condition is false.  **Returned Information:** The output parameter status.  **Exit state:** Entry state. |
| maxLogs\_reached | **Condition:** Pre-condition is true AND post-condition is false.  **Returned Information:** The output parameter status.  **Exit state:** Entry state. |
| operation\_failed | **Condition:** Pre-condition is true AND post-condition is false.  **Returned Information:** The output parameter status.  **Exit state:** Entry state. |

### 6.3.2 Operation unsubscribeLog (M)

#### 6.3.2.1 Definition

Using this operation, the IRPManager that started a specific log is able to stop this log via unsubscribing. Note stopping a log implies that the log becomes invisible across Itf-N, independent from the loggingEndTime; therefore the IRPManager should retrieve log information of interest before using this unsubscribeLog operation, e.g. through the exportLogRecords operation.

The disposition of a log that has been stopped, that is, whether the log remains visible across the Itf-N, is left as vendor specific functionality. The time of the deletion of logs is vendor specific.

In consideration of a multi-manager environment, the log can only be stopped by the creating IRPManager if this IRPManager provided a related token while initiating subscribeLog.

#### 6.3.2.2 Input parameters

| Parameter Name | Qualifier | Information type | Comment |
| --- | --- | --- | --- |
| logSubscriptionId | M | Log.logSubscriptionId | See clause 5.5.1 |
| logManagerToken | O | Log.logManagerToken | See clause 5.5.1  If subscribeLog returned a logManagerToken for a log with logSubscriptionId=”X”, then logManagerToken must be provided as input parameter of this operation to successfully stop log with logSubscriptionId=”X”.  If subscribeLogdid not return a logManagerToken for a log with logSubscriptionId=”X”, then logManagerToken must contain NULL value to successfully stop log with logSubscriptionId=”X”. |

#### 6.3.2.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| status | M | ENUM (OperationSucceeded, OperationFailed) | If logSubscriptionId is valid and logManagerToken (if supported) is matching, status = OperationSucceeded.  If operation\_failed is true, status = OperationFailed. |

#### 6.3.2.4 Pre-condition

There are no pre-conditions, other than those established by the generic rules (see clause 6.2).

#### 6.3.2.5 Post-condition

logStopped

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logStopped | The specified log is stopped. |

#### 6.3.2.6 Exceptions

| Exception Name | Definition |
| --- | --- |
| operation\_failed | **Condition:** Pre-condition is true AND post-condition is false.  **Returned Information:** The output parameter status.  **Exit state:** Entry state. |

### 6.3.3 Operation exportLogRecords (O)

Support for this operation is mandatory if getLogRecords is not supported.

#### 6.3.3.1 Definition

Using this operation, an IRPManager can initiate export of all or part of a log into a file. This file then is being transferred to the IRPManager using the File Transfer IRP (note also that the FTIRP is providing the file location to the IRPManager).

#### 6.3.3.2 Input parameters

| Parameter Name | Qualifier | Information type | Comment |
| --- | --- | --- | --- |
| logSubscriptionId | M | Log.logSubscriptionId | See clause 5.5.1 |
| notificationCategories | O | Log.notificationCategories | See clause 5.5.1 |
| filter | O | Log.filterLog.filter | See clause 5.5.1 |

#### 6.3.3.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| invocationId | M | -- | This parameter carries an identifier that NLIRP assigns to identify the request if the status is OperationSucceeded. This parameter carries no information if the status is not OperationSucceeded.  One identifier can at most identify one request. NLIRP should not reuse identifier values.  This identifier shall also appear in the file name of the exported log file as <specificIRP\_extension> (as part specified in Appendix A: File Name Convention of [8]).  When the exported log file is ready, FTIRP issues notifyFileReady notification which carries the exported log file name. |

#### 6.3.3.4 Pre-condition

There are no pre-conditions, other than those established by the generic rules (see clause 6.2).

#### 6.3.3.5 Post-condition

logRecordsExported

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logRecordsExported | The specified log records have been exported as requested. In case the log is empty or in case that all the log records do not satisfy the criteria of input parameters notificationCategories and filter, this post-condition is true. |

#### 6.3.3.6 Exceptions

| Exception Name | Definition |
| --- | --- |
| export\_failed | The NLIRP was unable to export the specified records. |
| operation\_failed | **Condition:** Pre-condition is true AND post-condition is false.  **Returned Information:** The output parameter status.  **Exit state:** Entry state. |

### 6.3.4 Operation getLogRecords (O)

Support for these operations is mandatory if exportLogRecords is not supported.

#### 6.3.4.1 Definition

Using this operation, an IRPManager can retrieve one or more log records from a certain log.

Note that this operation might be preferred for retrieval of small amounts of log records, while operation exportLogRecords might be preferred for retrieval of medium to large amounts of log records, as providing a more efficient bulk transfer mechanism.

#### 6.3.4.2 Input parameters

| Parameter Name | Qualifier | Information type | Comment |
| --- | --- | --- | --- |
| logSubscriptionId | M | Log.logSubscriptionId | See clause 5.5.1 |
| notificationCategories | O | Log.notificationCategories | See clause 5.5.1 |
| filter | O | Log.filter | See clause 5.5.1 |

#### 6.3.4.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| getLogRecordsResult | M | List of logRecord's | List of logRecords; each entry of the list holds all supported attributes of IOC LogRecord (see 5.3.3.2). |
| status | M | ENUM (OperationSucceeded, OperationFailed) | If logSubscriptionId is valid and (logRecordIdList is empty or logRecordIdList contains valid Id's), status = OperationSucceeded.  If operation\_failed is true, status = OperationFailed. |

#### 6.3.4.4 Pre-condition

There are no pre-conditions, other than those established by the generic rules (see clause 6.2).

#### 6.3.4.5 Post-condition

logRecordsRetrieved

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logRecordsRetrieved | The specified log records have been retrieved as requested. If the log is empty or all the log records do not satisfy the criteria of input parameters notificationCategories and filter, this post-condition is true. |

#### 6.3.4.6 Exceptions

| Exception Name | Definition |
| --- | --- |
| operation\_failed | **Condition:** Pre-condition is true AND post-condition is false.  **Returned Information:** The output parameter status.  **Exit state:** Entry state. |

## 6.4 NLIRPOperations\_2 Interface (O)

### 6.4.1 Operation getLogSubscriptionIds (M)

#### 6.4.1.1 Definition

Using this operation, an IRPManager can query the NLIRP for all available log subscriptions.

#### 6.4.1.2 Input parameters

| Parameter Name | Qualifier | Information type | Comment |
| --- | --- | --- | --- |
| -- | -- | -- | -- |

#### 6.4.1.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| logSubscriptionIds | M | List of logSubscriptionId's | See clause 5.5.1 & 6.4.1.2.  Note that empty list is a valid value. |
| status | M | ENUM (OperationSucceeded, OperationFailed) | If operation is successful, status = OperationSucceeded.  If operation\_failed is true, status = OperationFailed. |

#### 6.4.1.4 Pre-condition

There are no pre-conditions, other than those established by the generic rules (see clause 6.2).

#### 6.4.1.5 Post-condition

There are no post-conditions. Querying of log subscription Id’s does not result in any changes within the IRP Agent.

#### 6.4.1.6 Exceptions

| Exception Name | Definition |
| --- | --- |
| operation\_failed | **Condition:** Pre-condition is true AND post-condition is false.  **Returned Information:** The output parameter status.  **Exit state:** Entry state. |

### 6.4.2 Operation getLogSubscriptionStatus (M)

#### 6.4.2.1 Definition

Using this operation, an IRPManager can query the NLIRP for available log status information of an individual log.

#### 6.4.2.2 Input parameters

| Parameter Name | Qualifier | Information type | Comment |
| --- | --- | --- | --- |
| logSubscriptionId | M | Log.logSubscriptionId | See clause 5.5.1 |

#### 6.4.2.3 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| logAttributeList | M | Attributes of related Log IOC except logManagerToken | List of all supported attributes of IOC Log except logManagerToken (see clause 5.3.2.2). |
| status | M | ENUM (OperationSucceeded, OperationFailed) | If logSubscriptionId is valid status = OperationSucceeded.  If operation\_failed is true, status = OperationFailed. |

#### 6.4.2.4 Pre-condition

There are no pre-conditions, other than those established by the generic rules (see clause 6.2).

#### 6.4.2.5 Post-condition

There are no post-conditions. Querying of log attributes does not result in any changes within the IRPAgent.

#### 6.4.2.6 Exceptions

| Exception Name | Definition |
| --- | --- |
| operation\_failed | **Condition:** Pre-condition is true AND post-condition is false.  **Returned Information:** The output parameter status.  **Exit state:** Entry state. |

## 6.5 NLIRPNotifications\_1 Interface (M)

### 6.5.1 Notification notifyLogSubscribed (M)

#### 6.5.1.1 Definition

Using this notification, an IRPAgent informs all subscribed IRPManagers that a log subscription has been successfully initiated.

#### 6.5.1.2 Input Parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| objectClass | M,Y | Log.objectClass | Notification header - see [3]. It shall carry the Log class name. |
| objectInstance | M,Y | Log.objectInstance | Notification header - see [3]. It shall carry the DN of Log. |
| notificationId | M,N | -- | Notification header - see [3]. |
| eventTime | M,Y | -- | Notification header - see [3]. |
| notificationType | M,Y | "notifyLogSubscribed" | Notification header - see [3]. |
| systemDN | C,Y | -- | Notification header - see [3]. |
| logSubscriptionId | M,N | Log.logSubscriptionId | See clause 5.5.1 |
| loggingEndTime | O,N | Log.loggingEndTime | See clause 5.5.1 |
| notificationCategories | O,N | Log.notificationCategories | See clause 5.5.1; if absent than all notifications are being logged |
| filter | O,N | Log.filter | See clause 5.5.1 |

#### 6.5.1.3 Triggering Event

##### 6.5.1.3.1 From-state

subscribeLog

|  |  |
| --- | --- |
| Assertion Name | Definition |
| subscribeLog | An IRPManager requests that a new log be started. |

##### 6.5.1.3.2 To-state

logStarted

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logStarted | The NLIRP has started the requested logging activity. |

### 6.5.2 Notification notifyLogUnsubscribed (M)

#### 6.5.2.1 Definition

Using this notification, an IRPAgent informs all subscribed IRPManager that a log subscription has stopped.

#### 6.5.2.2 Input Parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| objectClass | M,Y | Log.objectClass | See table 6.5.1.2. |
| objectInstance | M,Y | Log.objectInstance | See table 6.5.1.2. |
| notificationId | M,N | -- | See table 6.5.1.2. |
| eventTime | M,Y | -- | See table 6.5.1.2. |
| notificationType | M,Y | "notifyLogUnsubscribed" | See table 6.5.1.2. |
| systemDN | C,Y | --. | See table 6.5.1.2. |
| logSubscriptionId | M,N | Log.logSubscriptionId | See table 6.5.1.2. |

#### 6.5.2.3 Triggering Event

##### 6.5.2.3.1 From-state

unsubscribeLog OR loggingEndTimeReached

|  |  |
| --- | --- |
| Assertion Name | Definition |
| unsubscribeLog | The IRPManager that started the log requests that the log be unsubscribed. |
| loggingEndTimeReached | The logging end timespecified for the log in subscribeLog has been reached. |

##### 6.5.2.3.2 To-state

logStopped

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logStopped | The logging activity has stopped. |

## 6.6 NLIRPNotifications\_2 Interface (O)

### 6.6.1 Notification notifyOccupancyLevelCrossed (M)

#### 6.6.1.1 Definition

Using this notification, an IRPAgent informs all subscribed IRPManagers about the occupancy level of a certain log according with levels defined in 5.5.1 (in addition also the log full action type of this NLIRP instance is being provided for context purposes).

#### 6.6.1.2 Input Parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| objectClass | M,Y | Log.objectClass | See table 6.5.1.2. |
| objectInstance | M,Y | Log.objectInstance | See table 6.5.1.2. |
| notificationId | M,N | -- | See table 6.5.1.2. |
| eventTime | M,Y | -- | See table 6.5.1.2. |
| notificationType | M,Y | "notifyOccupancyLevelCrossed" | See table 6.5.1.2. |
| systemDN | C,Y | -- | See table 6.5.1.2. |
| logSubscriptionId | M,Y | Log.logSubscriptionId | See table 6.5.1.2. |
| currentOccupancyLevel | M,N | The integer portion of (( Log.currentSize/ Log.maxSize)\*100) | See clause 5.5.1. |
| logFullAction | O,N | Log.logFullAction | See clause 5.5.1. |

#### 6.6.1.3 Triggering Event

##### 6.6.1.3.1 From-state

logFull OR occupancyLevelCrossed

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logFull | The log is full, that is, the number of log records contained within the log has exceeded the maximum number of log records that was established at log startup (log capacity has reached 100 %). |
| occupancyLevelCrossed | The number of log records within the log has crossed one of the threshold boundaries (see 5.5.1 Log.occupancyLevels). |

##### 6.6.1.3.2 To-state

logWrapping OR (logHalted OR logging)

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logWrapping | Applicable for case Log.logFullAction=’wrap’: The number of log records within the log has exceeded the maximum number of allowed log records (100 % capacity has been crossed). The logging of new records will cause the oldest records within the log to be deleted and replaced (first in, first out). The deletion of records may occur in a block, such that the log transitions back to the “started” state prior to any new records being written. |
| logHalted | Applicable for case Log.logFullAction=’halt’: The number of log records within the log has exceeded the maximum number of allowed log records (100 % capacity has been crossed). The logging of new records will be discontinued until the deletion of records has occurred (deletion of records is outside the scope of this IRP). |
| logging | Applicable for case Log.logFullAction=’halt’: The number of log records within the log is below the maximum number of allowed records. The generation of the notification is done to inform the subscribed IRPManagers that the log is filling up. |

### 6.6.2 Notification notifyLoggingResumed (O)

#### 6.6.2.1 Definition

Using this notification, an IRPAgent informs all subscribed IRPManagers that the amount of data within a given log has been reduced, allowing logging to resume continue (according to the information provided when the log was created). Note that this notification only applies to Log.logFullAction == ‘halt’.

#### 6.6.2.2 Input Parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| objectClass | M,Y | Log.objectClass | See table 6.5.1.2. |
| objectInstance | M,Y | Log.objectInstance | See table 6.5.1.2. |
| notificationId | M,N | -- | See table 6.5.1.2. |
| eventTime | M,Y | -- | See table 6.5.1.2. |
| notificationType | M,Y | "notifyLoggingResumed" | See table 6.5.1.2. |
| systemDN | C,Y | -- | See table 6.5.1.2. |
| logSubscriptionId | M,Y | Log.logSubscriptionId | See table 6.5.1.2. |

#### 6.6.2.3 Triggering Event

##### 6.6.2.3.1 From-state

logFull

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logHalted | The log is full, that is, the number of log records contained within the log has exceeded the maximum number of log records that was established at log startup (log capacity has reached 100 % and Log.logFullAction=’halt’). |
| occupancyLevelCrossed | The number of log records within the log has crossed one of the threshold boundaries (see 5.5.1 Log.occupancyLevels). |

##### 6.6.2.3.2 To-state

|  |  |
| --- | --- |
| Assertion Name | Definition |
| logging | The number of log records within the log is below the maximum number of allowed records. The generation of the notification is done to inform the subscribed IRPManagers that the log is filling up. |

Annex A (informative):  
Change history

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change history | | | | | | | | |
| Date | TSG # | TSG Doc. | CR | Rev | Subject/Comment | Cat | Old | New |
| Mar 2004 | S\_23 | SP-040123 | -- | -- | Submitted to SA#23 for Information | -- | 1.0.0 |  |
| Dec 2004 | S\_26 | SP-040798 | -- | -- | Submitted to SA#26 for Approval | -- | 2.0.0 | 6.0.0 |
| Jun 2005 | S\_28 | SP-050290 | 0001 | -- | Correct mapping info of currentOccupancyLevel and some editorial errors | F | 6.0.0 | 6.1.0 |
| Jun 2005 | S\_28 | SP-050329 | 0002 | -- | Apply Generic System Context – Align with TS 32.150 | F | 6.0.0 | 6.1.0 |
| Mar 2006 | SA\_31 | SP-060091 | 0003 | -- | Add invocationId output parameter of exportLogRecords operation | F | 6.1.0 | 6.2.0 |
| Mar 2006 | SA\_31 | SP-060089 | 0004 | -- | Correct ambiguity of object class and object instance usage in notification | F | 6.1.0 | 6.2.0 |
| Dec 2006 | SA\_34 | SP-060709 | 0005 | -- | Correct the errors in the definition of occupancyLevels and currentOccupancyLevel | F | 6.2.0 | 6.3.0 |
| Jun 2007 | SA\_36 | -- | -- | -- | Automatic upgrade to Rel-7 (no CR) at freeze of Rel-7. Deleted reference to CMIP SS, discontinued from R7 onwards. | -- | 6.3.0 | 7.0.0 |
| Sep 2007 | SA\_37 | SP-070614 | 0006 | -- | Remove the use of visibility symbol | C | 7.0.0 | 8.0.0 |
| Dec 2009 | - | - | - | - | Update to Rel-9 version (MCC) | - | 8.0.0 | 9.0.0 |
| Mar 2011 | - | - | - | - | Update to Rel-10 version (MCC) | - | 9.0.0 | 10.0.0 |
| 2012-09 | - | - | - | - | Update to Rel-11 version (MCC) |  | 10.0.0 | **11.0.0** |
| 2014-10 | - | - | - | - | Update to Rel-12 version (MCC) |  | 11.0.0 | **12.0.0** |
| 2016-01 | - | - | - | - | Update to Rel-13 version (MCC) |  | 12.0.0 | **13.0.0** |
| 2017-03 | SA#75 | - | - | - | Promotion to Release 14 without technical change |  | 13.0.0 | **14.0.0** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2018-06 |  |  |  |  |  | Update to Rel-15 version (MCC) | 15.0.0 |
| 2020-07 | - | - | - | - | - | Update to Rel-16 version (MCC) | **16.0.0** |