

# MTAS Barring and Dial Plan Services Management Guide

MTAS

USER GUIDE

**Copyright**

© Ericsson AB 2016–2019. 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Prerequisites	1
<b>2</b>	<b>Overview</b>	<b>3</b>
2.1	Subfunctions	5
2.2	CB Interaction with Other Services	7
2.3	Traffic View	11
2.4	Configuration View	12
2.5	Identity Conditions in Barring Services	14
<b>3</b>	<b>Service Invocation</b>	<b>15</b>
3.1	Anonymous Condition Checks	19
3.2	Roaming Condition for Mobile User Checks	20
3.3	International Condition for Mobile User Checks	21
3.4	International-exHC Condition for Mobile User Checks	21
3.5	Roaming, International, and International-exHC for Mobile User on Non-3GPP Access	21
3.6	Checking for Call Analysis Condition	22
<b>4</b>	<b>Barring Rules</b>	<b>23</b>
4.1	Relationship between OCB Types and Dial Plans	23
4.2	Relationship between ICB Types	29
4.3	Global Barring Rules	32
4.4	Global White List	33
4.5	Dial Plan	33
4.6	Location Based Barring	34
4.7	Relationship between Barring Programs and User Barring Categories	34
4.8	Relationship between Operator Barring Programs, Operator Permitted Programs, and Operator Diversion Barring Programs	39
4.9	Subscription Rules	41
4.10	Evaluation of Rules	50
<b>5</b>	<b>CB Service Configuration</b>	<b>53</b>
5.1	Overview Tables and Activation	53



5.2	Configure ACR Display Name Evaluation	56
5.3	Configure a User Barring Category	56
5.4	Configure an Operator Barring Category	59
5.5	Configure Localness Barring Categories	63
5.6	Configure Location Based Barring	64
5.7	Configure a Barring Program in Single Scheme	66
5.8	Configure a Barring Program in Multiple Scheme	69
5.9	Configure the Global White List	70
5.10	Configure a Dynamic Black List	73
5.11	Configure an Announcement	73
5.12	Configure SIP Error Response Codes from MTAS	73
5.13	Configure a Cause Value	74
5.14	Configure the Communication Barring Administrative State	74
5.15	Configure the Dial Plan Administrative State	74
5.16	Configure Wholesale for Communication Barring	74
5.17	Configure the Nodal Dial Plan	75
5.18	Configure the Per-VTP Dial Plan	77
5.19	Configure Service Data	81
<b>6</b>	<b>Performance Management</b>	<b>87</b>
<b>7</b>	<b>Fault Management</b>	<b>89</b>
<b>8</b>	<b>Barring Rule Examples</b>	<b>91</b>
8.1	Bar Incoming Communication from Alice	91
8.2	Bar Incoming Communication from Recent Caller with Privacy	92
8.3	Bar Incoming Communication from Recent Caller without Privacy	93
8.4	Bar Incoming Communication from Anonymous	94
8.5	Bar Incoming Communication from example.com except from Alice and Bob	95
8.6	Bar Incoming from Range of Numbers	95
8.7	Bar Incoming Communication Based on Served Identity	96
8.8	Bar Outgoing Communication Based on Served Identity	96
8.9	Bar Outgoing Communication Based on Subscriber-state Except 1800 Pre-Paid Calls	97
8.10	Bar Incoming Communication Based on Subscriber-state	98
8.11	White List	99



8.12	Conditions Combined in One Rule and Different Rules	100
8.13	Playing Generic Announcement – play-announcement	103
8.14	Playing Generic Announcement – play-segmented-announcement	104
8.15	Do Not Disturb Communication Barring	104
8.16	Bar Outgoing Communication Based on Call Analysis	105
<b>9</b>	<b>OCB Barred and Exempted List Configuration Examples</b>	<b>107</b>
9.1	Wildcard Pattern in List Entry	107
9.2	Fixed Length Pattern in List Entry	110





# 1 Introduction

This document describes how to configure the Communication Barring (CB) service, including several Barring services, in the MTAS.

In addition, this document describes how to configure the related Dial Plan service, in the MTAS. The Dial Plan function is described as part of the CB service.

## 1.1 Prerequisites

It is assumed that the user of this document is familiar with the O&M area, in general.

### 1.1.1 Licenses

To enable basic services in the MTAS, the MMTel AS Voice Base license must be installed.

For more information about the MMTel licenses, see [MTAS licenses](#).

### 1.1.2 Documents

Before starting any procedure in this document, ensure that the following documents are available:

- [Ericsson Command-Line Interface User Guide](#)
- [Managed Object Model \(MOM\)](#)

### 1.1.3 Conditions

- The following condition must apply:

An Ericsson Command-Line Interface (ECLI) session in Exec mode is in progress.







## 2 Overview

The MTAS offers several Barring services, which are controlled by the interaction between the Managed Objects (MOs) and per subscriber data. The Barring services make it possible for a subscriber to use barring of certain categories of communication.

The CB service interacts with other services and is colocated with other simulation services on the MTAS, for example, Communication Forwarding, Hold, and Ad hoc Conference.

The function is executed both at the originating and terminating MTAS. For an initial INVITE, the subfunction Outgoing Communication Barring (OCB) is executed on the originating MTAS and the subfunction Incoming Communication Barring (ICB) is executed on the terminating MTAS. The OCB can also execute on the terminating side if, for example, the communication is diverted.

For a REFER, the subfunction OCB is executed on the MTAS of the REFER sender; currently MTAS only supports sending of a REFER by a Conference Creator. For an INVITE caused by a REFER, the subfunction OCB is executed on the original originating MTAS, and the subfunctions OCB and ICB are executed on the original terminating MTAS.

The ICB and OCB are triggered for both registered and unregistered users when communication is diverted. There are no differences in the execution because of the registration state.

The Anonymous Communication Rejection (ACR) service is a kind of ICB, which bars anonymous callers. The ACR is considered to be rule-based barring.

Do Not Disturb Communication Barring (DNDCB) is a kind of ICB as the barring part of Do Not Disturb (DND) service. DND is a commonly used expression for indicating a condition where the served user does not wish to be interrupted by incoming calls. Incoming calls encountering a DND condition at the terminating party is to be barred. Like ACR, DNDCB is also considered to be rule-based barring.

The CB service determines to bar or allow the communication by evaluating barring rules. In addition to explicit blacklists (only with the condition identity), explicit global white lists (only with the condition identity), dial plans (only with the condition identity), and Barring Programs (only with the condition identity), the served user can, for example, do the following using the CB service:

- Bar incoming anonymous communication
- Bar incoming communication because of Do Not Disturb activation
- Bar incoming communication when the served mobile user is roaming
- Bar incoming or outgoing communication based on the served identity (primary ID or alias) of the user



- Bar outgoing communication to a specific identity during a time period
- Bar all outgoing communication except to some identities (for example, white list)
- Bar outgoing communication when the served mobile user is roaming
- Bar outgoing communication from international calls
- Bar outgoing communication from international-exHC calls
- Bar outgoing communication based on call analysis

The rules are built up with different conditions, which can be combined in many ways to express when a communication is to be barred. The different conditions are as follows:

- Identity (the identity can match single user, domains, and so on)
- Anonymous
- Validity (time duration)
- Media (the communication contains the specified media type)
- Communication-diverted (the communication has previously been diverted)
- Rule-deactivated (makes it possible to disable a rule without removing it)
- Other-identity (all identities that have not been matched in other rules)
- Roaming (for mobile user)
- International (for mobile user)
- International-exHC (for mobile user)
- Served-identity
- `b-number-type` (obtained from Request-URI in Number Translation)
- `b-network-type` (obtained from NRN mapped to configured network type (for example, Mobile or Fixed) in Number Portability)
- `localness` (obtained from Number Analysis; that is, Local, LongDistance or International)

In addition, the identity of a recent caller can be added to the ICB ruleset, and the outgoing Barring Programs applicable to a user can be changed by the user using Supplementary Service Codes.

For more information about the Supplementary Service Codes, see [MTAS Supplementary Service Codes Management Guide](#).



When communication is diverted and Application Server (AS) chaining is disabled, originating services like OCB are executed directly in terminating MTAS.

When communication is diverted and AS chaining is enabled, the INVITE is returned to Serving Call Session Control Function (S-CSCF) after retargeting. Invocation of originating services like OCB is triggered by S-CSCF by sending the INVITE to the terminating MTAS (or other AS) for the originating session case.

For more information about Originating AS chaining, see [MTAS SIP Management Guide](#).

## 2.1 Subfunctions

This section describes the subfunctions included in the CB service.

### 2.1.1 Incoming Communication Barring

The ICB subfunction makes it possible for a user to use barring of certain categories of incoming communications according to the global ICB blacklist, and the barring rules of the user.

ICB rejects incoming communication when the evaluation of the Incoming Communication Barring rules of the served user evaluates to (allow=false). ICB does not bar communication from calling parties which match with the ICB global white list.

The incoming communication is rejected by a SIP response with the result code defined in the MO `mtasIcbRejectCode`, default is 603 (Decline).

Configurable MOs and attributes related to the CB services are defined in [Managed Object Model \(MOM\)](#).

### 2.1.2 Anonymous Communication Rejection

The Anonymous Communication Rejection (ACR) subfunction allows the served user to reject incoming communications on which the Public User Identity of the originating user is restricted.

For example, the OIR restricts the presentation of the originating user's Public User Identity and that triggers ACR in the terminating MTAS.

The incoming communication is rejected by a SIP response with result code 433 (Anonymity disallowed). The originating user is given an appropriate indication that the communication has been rejected because of the ACR function.

The ACR subfunction is a special case of the ICB function and is a regulatory service in many countries.



### 2.1.3 Do Not Disturb Communication Barring

The DNDCB subfunction allows the served user to reject incoming communications when DNDCB service for the user is activated.

For example, in the case where the served user does not wish to be interrupted by any incoming communication, the served user can activate DNDCB with any condition type to bar the incoming communication.

The incoming communication is rejected by a SIP response with result code 480 (Temporarily Unavailable). The originating user is given an appropriate indication that the communication has been rejected because of the DNDCB function.

### 2.1.4 Outgoing Communication Barring

The OCB subfunction makes it possible for a user to use barring of certain categories of outgoing communications according to the global OCB blacklist, the diversion global blacklist, the Location Based Barring, the user's Operator Barring Program, Operator Permitted Program, Operator Diversion Barring Program, user Barring Programs and barring rules.

OCB rejects outgoing communications when the evaluation of the served user's Operator Barring Program, Operator Permitted Program, Operator Diversion Barring Program, user Barring Programs, or Outgoing Communication Barring rules evaluates to (`allow=false`). OCB does not bar Communication to called parties which match with the OCB global white list except for diverted calls where the diversion global blacklist takes precedence.

The outgoing communication is either rejected by a SIP response with result code 603 (Decline) or answered by 200 OK.

The latter happens if `mtas0cbPlayEarlyMedia` is set to 0 (disabled) and the Communication Barring service is configured for playing standard CB announcement or the matched barring rule is requesting to play a generic (segmented) announcement when the generic (segmented) announcement is configured.

### 2.1.5 Play Announcement

The Play announcement subfunction plays an announcement for the caller in addition to the result code in the final response.

There are two different ways of playing announcement:

- Playing standard announcement:

- Playing announcement by looking up the announcement code configured in the CM attribute.

- Playing generic announcement:



Playing the operator-named announcement indicated either in the play-announcement or the play-segmented-announcement action element within the matched OCB or ICB rule by using Generic Announcement service.

This operator-named announcement feature is applicable for all types of Communication Barring service, such as OCB, ICB, ACR, and DNDCB. Thus, the play-announcement action element can be inserted into any rule in the Communication Barring operator or subscriber subscription rule-set.

For more information about announcement handling and attributes for the CB services, refer to the following documents:

- MTAS Announcement Management Guide
- MTAS Generic Announcement Management Guide

#### 2.1.5.1 Announcement on Early Media or Established Sessions

The attribute `mtas0cbPlayEarlyMedia` defines if the OCB service is to play announcements on established sessions or early announcement on not established sessions. The default is to play announcement on an early media.

The following attributes determine if an announcement is to be played on an established session or as an early media:

- `mtasAcrPlayEarlyMedia`
- `mtasIcbPlayEarlyMedia`
- `mtasIcbDndPlayEarlyMedia`
- `mtas0cbPlayEarlyMedia`

Each of the parameters defines the way of playing announcements for the corresponding service. Also, the following parameters determine if an announcement is to be played on an established session:

- `mtasAcrEstablishedAnnRules`
- `mtasIcbEstablishedAnnRules`
- `mtasIcbDndEstablishedAnnRules`

By default, an announcement is played as an early media on a not established session.

## 2.2 CB Interaction with Other Services

This section describes the CB interaction with other services.



### 2.2.1 Ad hoc Conference

The CB interacts with the Ad hoc conference service by inspecting the headers Refer-To and Referred-By.

For more information about the Ad hoc conference service, see [MTAS Ad-hoc Conference Management Guide](#).

### 2.2.2 Address Policing

For more information about the interaction between the Barring and the Address Policing services, see [MTAS Address Policing Management Guide](#).

### 2.2.3 Call Admission Control

The ICB service processes the initial INVITE before the Call Admission Control (CAC) services.

The OCB service processes the initial INVITE after the CAC services.

For more information about the CAC service, see [MTAS Call Admission Control Management Guide](#).

### 2.2.4 Carrier Select Rn and Carrier Pre-Select Rn

The original domain name in the incoming request is only overwritten with the domain name of the Carrier, after the OCB service processing has been done.

For more information about the Carrier Select (CS) Rn and Carrier Pre-Select (CPS) Rn services, see [MTAS Carrier Select and Carrier Pre-Select Management Guide](#).

### 2.2.5 Charging

INVITE requests that are processed by the CB result in the generation of charging messages. The charging messages include a service-specific AVP identifying the type of barring, for example, ICB. If a communication is allowed or barred because of a matching ICB, OCB, ACR, or DNDCB rule, the identifier of the matched CB rule is also included in an extra service-specific AVP.

If the call was rejected by an Operator Barring Program, an Operator Diversion Barring Program, or a user Barring Program, then the value in the AVP is the one associated with the Operator Barring Category, user Barring Category, or special Barring Category that caused the call to be barred.

When OCB applies to a diverted INVITE, the service-specific AVP is only included in the charging message generated for the diverted (B-to-C) leg, for example, it is not included in the charging message generated for the incoming (A-to-B) leg.



Charging messages are not generated when REFER requests are rejected.

For more information about the Charging service, see [MTAS Charging Management Guide](#).

## 2.2.6 CLIR Interworking

When CLIR Interworking service is unlocked and if the calling party has OIR active, then those subscriber defined ICB rules which use the identity of the caller are not evaluated.

For more information about the interaction between the Barring and the CLIR services, see [MTAS CLIR Interworking Management Guide](#).

## 2.2.7 Communication Completion

ICB takes precedence over the Communication Completion (CC) services. If a call is rejected by ICB, the CC is not offered in the response sent to the caller.

For the CC Recall INVITE to the CC Caller OCB is not applied. For the CC Call to the CC Called Party, normal OCB checks are applied by the originating MTAS and normal ICB checks are applied by the terminating MTAS.

For more information about the CC services, see [MTAS Communication Completion Management Guide](#).

## 2.2.8 Communication Diversion

The OCB interacts with the Communication Diversion (CDIV) service by inspecting the INVITE destined for the diverted-to party before it is sent by the MTAS and is rejecting the communication with 486 Busy Here if it is not allowed.

The ICB also uses the History-Info header inserted by CDIV to check whether an incoming INVITE has already been diverted, to evaluate the communication-diverted condition. For more information about the CDIV service, see [MTAS Communication Diversion Management Guide](#).

CB for the diverting served user in a transit MTAS, OCB is an originating service that is either executed internally for a terminating session case or is started by S-CSCF to be executed for an originating session case.

While evaluating Roaming, International or International-exHC barring rules in transit AS, `mtasCbLocationInTransitMode` CM attribute provides optional configuration to set current country as home country for served user. If the CM parameter `mtasCbLocationInTransitMode` is set to 1, then MMTel AS in transit mode sets current country to home country before evaluating barring rules for served user.



For more information about Originating AS chaining, see [MTAS SIP Management Guide](#).

### **2.2.9 Communication Waiting**

ICB, ACR, and DNDCB take precedence over CW. If a subscriber has CW active and also ICB, ACR or DNDCB active, then the action of ICB, ACR, or DNDCB can result in the terminating communication attempt being rejected and CW not being started.

For more information about the CW service, see [MTAS Communication Waiting Management Guide](#).

### **2.2.10 Explicit Communication Transfer**

For more information about the interaction between the Barring and the ECT services, see [MTAS Explicit Communication Transfer Management Guide](#).

### **2.2.11 Flexible Identity Presentation**

The ICB service processes the initial INVITE in which the “From” and P-Asserted-Identity headers have been replaced by the FIP service. This means that the ICB service can apply an action on the call because of the FIP identity which would otherwise have not been applied, or do not apply an action on the call because of the FIP identity which has been applied otherwise.

For more information about the Identity Presentation services, see [MTAS Identity Presentation Management Guide](#).

### **2.2.12 Hotline**

The OCB is performed on the hotline destination before the outgoing INVITE. For more information about the Hotline service, see [MTAS Hotline Service Management Guide](#).

### **2.2.13 Malicious Communication Identification**

The communication details associated with the latest incoming communication, associated with the Malicious Communication Identification (MCID) service, are not updated when a communication request is rejected because of ICB, ACR, or DNDCB.

For more information about the MCID service, see [MTAS Malicious Communication Identification Management Guide](#).





### 2.2.14 Originating Identification Restriction Override

The Originating Identification Restriction (OIR) Override service takes precedence over the ACR service. If the served user has the OIR Override service, no incoming request is treated as anonymous, even if the `mtasAcrGlobal` attribute is set to Enabled.

For more information about the Identity Presentation services, see [MTAS Identity Presentation Management Guide](#).

### 2.2.15 Number Translation

Number Translation Service [49] determines the B-Number type from the Request URI, and CB Service uses it in call analysis for evaluating the OCB condition `mmt-serv:b-number-type`.

### 2.2.16 Number Portability

Number Portability Service [50] determines the B-Number subscription network type from the Network Routing Number (NRN) mapped to the configured network type, and CB Service uses it in call analysis for evaluating the OCB condition `mmt-serv:b-network-type`.

## 2.3 Traffic View

The CB performs the following steps which apply to all the Barring services:

1. Service invocation – an event triggers the execution of the CB, for example, incoming INVITE, see Section 3 on page 15.
2. Rules evaluation – the CB evaluates the rule set of the served user and determines if the communication is barred, see Section 4 on page 23.
3. Send indication – if the communication is barred, an indication is sent to the originating user. The indication is a final SIP response, optionally preceded by playing an announcement.

The traffic view of the CB is shown in Figure 1. The subfunctions of the CB are triggered by a SIP event, for example, an outgoing INVITE triggers OCB and an incoming INVITE triggers ICB, ACR or DNDCB.

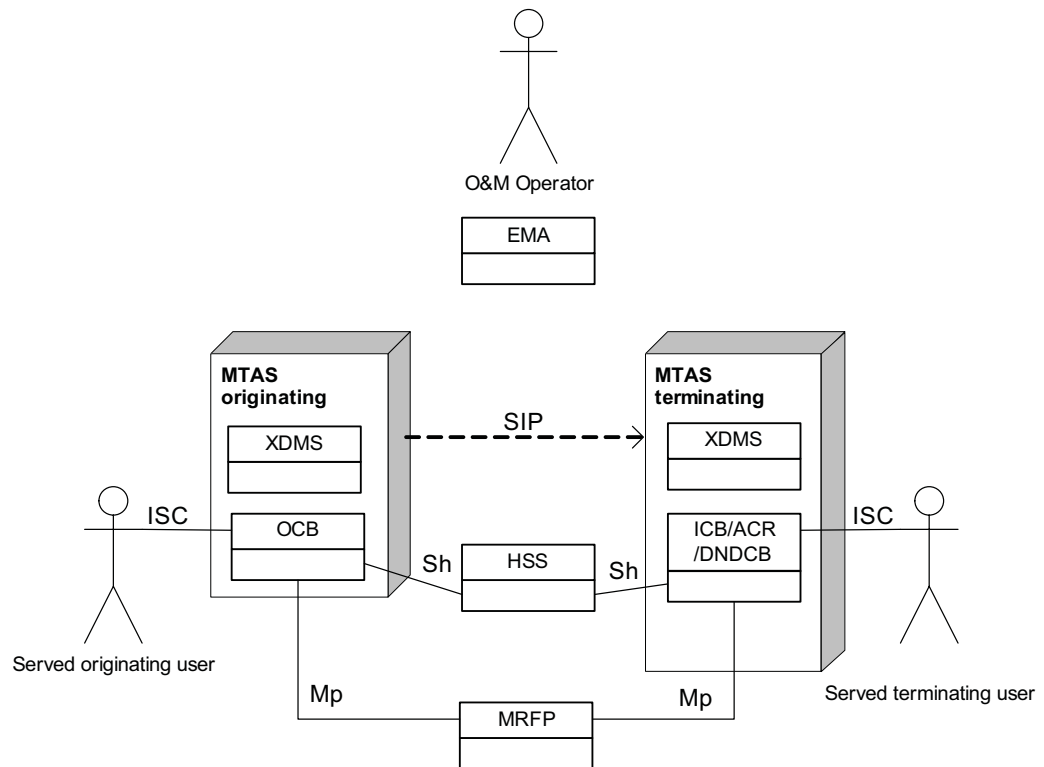


Figure 1 Traffic View of CB

## 2.4 Configuration View

The configuration view of the CB is shown in Figure 2. The following two main categories of configuration exist:

- CB on node level, for example, Lock/Unlock.
- Global barring rules, see Section 4 on page 23.

In node level configuration, the operator customizes the CB service by, for example, defining if announcement is used as indication also to the SIP response (603 Decline) and the announcement code, which defines the announcement to be played, defining the white lists, defining the Dial Plans, defining the Barring Categories.

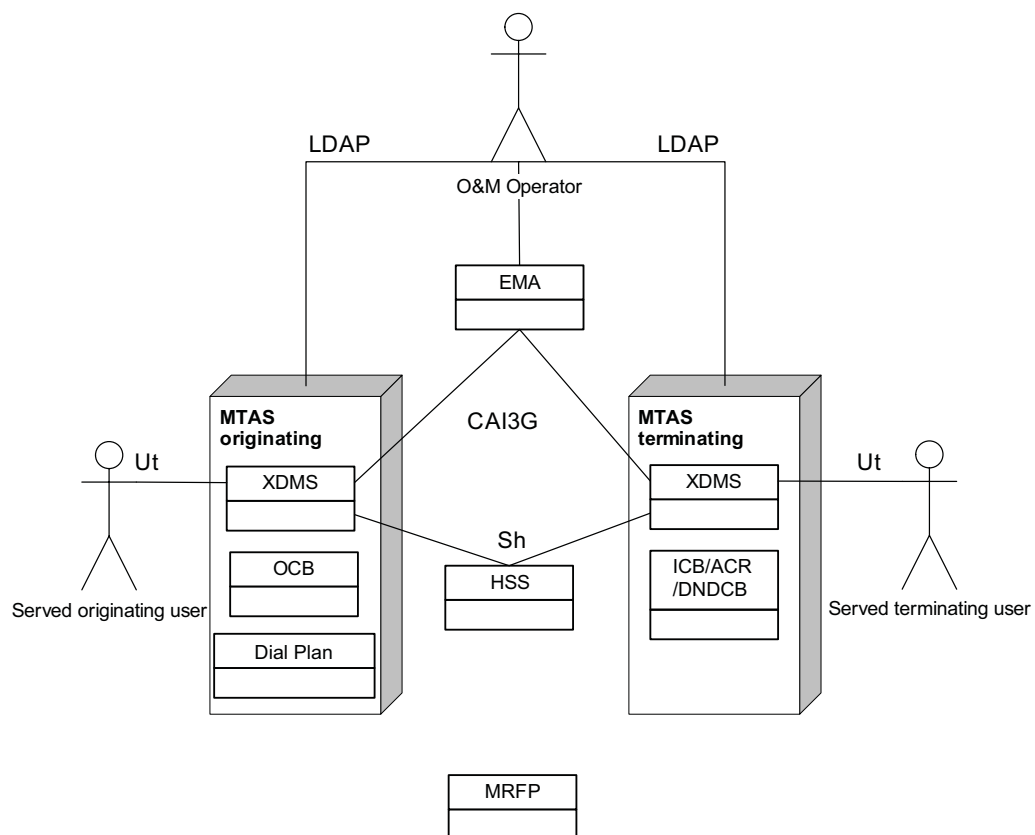


Figure 2 Configuration View of CB

The global barring rules are managed by the operator and consist of an incoming blacklist, an outgoing blacklist, an incoming white list, an outgoing white list (each of which only includes the condition identity) and reject anonymous (enable/disable).

The Diversion Global blacklist is managed by the operator and consists of an outgoing blacklist that only applies to diverted calls, and which only includes the condition identity.

The Dial Plan Consists of a nodal Dial Plan, managed by the OTP operator, an OTP-controlled per-VTP Dial Plan, managed by the OTP operator, and a VTP-controlled per-VTP Dial Plan, managed by the Virtual Telephony Provider (VTP) operator, each of which only includes the condition identity.

The Operator Barring Programs are configured by the operator, managed through XML Document Management Server (XDMS) that provides the CAI3G interface to the operator.

The Operator Diversion Barring Programs are configured by the operator, managed through XDMS that provides the CAI3G interface to the operator.

The Operator Permitted Programs are managed through XDMS that provides the CAI3G interface to the operator.



The Barring Programs are configured by the operator, managed through XDMS that provides the Ut interface (XCAP over HTTP) to the served user and the CAI3G interface to the operator, and managed through SSCs.

The subscription rules are managed through XDMS that provides the Ut interface (XCAP over HTTP) to the served user and the CAI3G interface to the operator, and (for ICB) managed through SSCs.

XDMS uses Sh (Diameter) to update the Home Subscriber Server (HSS).

The Location Based Barring is managed by the operator and consists of a list of number prefixes to be barred from a specific location area and a list of prefixes which are exempted from barring.

The relationships between different types of OCB are described in Section 4.1 Relationship between OCB Types and Dial Plans on page 23 and the different types of ICB are described in Section 4.2 Relationship between ICB Types on page 29.

## 2.5 Identity Conditions in Barring Services

The URIs `sip:+4981770124@example.com;user=phone` and `sip:+4981770124@example.com` are different addresses. These addresses can either be related to the same user or not. The inclusion of the `user=phone` parameter indicates that the URI is a tel URI that has been converted to a SIP URI in accordance with section 19.1.6 in [IETF RFC 3261](#).

The URI without a `user=phone` parameter indicates that it is a standard SIP name address of another possible user.



### 3 Service Invocation

The CB is started for the served user by various events. The subfunctions are started on originating, diverting, and terminating MTAS, for both registered and unregistered users, as shown in Table 1.

The diverting MTAS acts as both terminating and originating MTAS.

Table 1 Invocation of Subfunctions

Subfunction	MTAS Type	User	
		Registered	Unregistered
OCB	Originating Diverting	X	X
ICB	Diverting Terminating	X	X

Service invocation means start evaluating OCB or ICB rules; if a rule evaluates to `allow=false` and no rule evaluates to `allow=true`, then the communication is barred. This means that only starting CB does not mean that a communication is barred, but CB checks if it is to be barred.

The events that start the CB are listed in Table 2. Throughout Table 2, wherever the identity condition is checked as part of ICB, if the anonymous condition is displayed in a rule, a check for anonymity is performed in accordance with the service invocation and barring rules, as shown in Figure 3. When an action is given as “none”, the CB ignores a message that is expected to be checked for barring.

Throughout the table, if the roaming, the international and the international-exHC conditions appear in a rule, checks for these conditions are performed in accordance with the rules listed in Section 3.2 Roaming Condition for Mobile User Checks on page 20, Section 3.3 International Condition for Mobile User Checks on page 21, and Section 3.4 International-exHC Condition for Mobile User Checks on page 21 respectively.



Table 2 BC Service Start

Id	Request	MTAS Type	Served User Role	Function	Header	Action	Comment
1	INVITE	Originating	sender	OCB	Request-URI: B  P-Asserted-Identity: A  P-Access-Network-Info: A	Request-URI value, B, is matched with condition identity.  Refer-To: A Alternatively, the current location of the served user is checked with the home location and the home location of B.  If present, the P-Served-User otherwise the P-Asserted-Identity value is matched with condition served-identity.  The PANI-header (P-Access-Network-Info) is matched with Location Based Barring (OcbLb) rules.	This is OCB on the initial INVITE.  P-Served-User is used only if mtasSipSupportPServedUserHeader is set to 1.  PANI is used by operator to bar a call going to a specific destination depending on the mobile location the call was made from.
2	re-INVITE	Originating	sender	OCB	Request-URI: B  P-Asserted-Identity: A	None	This is a re-INVITE in an existing call, no need to check barring.
3	REFER	Originating	sender	OCB	Request-URI: B  Referred-By: A  Refer-To: C  P-Asserted-Identity: A	Refer-To value, C, is matched with condition identity.	
4	re-INVITE	Originating	receiver	OCB	Request-URI: A  P-Asserted-Identity: B	None	This is a re-INVITE in an existing call, no need to check barring.



Table 2 BC Service Start

Id	Request	MTAS Type	Served User Role	Function	Header	Action	Comment
5	re-INVITE	Terminating	sender	ICB	Request-URI: A P-Asserted-Identity: B	None	This is a re-INVITE in an existing call, no need to check barring.
6	INVITE	Terminating	receiver	ICB	Request-URI: B P-Asserted-Identity: A From: A	P-Asserted-Identity and From Header is matched with condition identity.  Alternatively, the current location of the served user is checked with the home location.	This is ICB on the initial INVITE. The From Header is used when checking ICB if the CM attribute mtasIcbUseFromHeader is Enabled.  When CLIR Interworking service is unlocked then if the calling party has Originating Identity Restriction active, then those subscriber defined ICB rules which use the identity of the caller are not evaluated.
7	INVITE	Terminating	receiver	ICB	Request-URI: B Referred-By: A P-Asserted-Identity: A From: A	Referred-By value, A, is matched with the condition identity.  If present, the P-Served-User otherwise the Request URI is matched with condition served-identity.	: P-Served-User is used only if mtasSipSupportPServedUserHeader is set to 1.
8	re-INVITE	Terminating	receiver	ICB	Request-URI: B P-Asserted-Identity: A	If present, the P-Served-User otherwise the Request URI is matched with condition served-identity.	This is a re-INVITE in an existing call, no need to check barring.  P-Served-User is used only if mtasSipSupportPServedUserHeader is set to 1.
9	REFER	Terminating	receiver	ICB	Request-URI: B Referred-By: A Refer-To: AS-A P-Asserted-Identity: A	None	The caller in the original call is trying to transfer the called party (served user) to C.



Table 2 BC Service Start

Id	Request	MTAS Type	Served User Role	Function	Header	Action	Comment
10	INVITE	Transit Terminating	diverter	OCB	Request-URI: C P-Asserted-Identity: A	Request-URI value, C, is matched with condition identity.	This is when AS chaining is disabled, OCB is checked in terminating session case.
11	INVITE	Transit Originating	diverter	OCB	Request-URI: C P-Asserted-Identity: A P-Served-User: B	Request-URI value, C, is matched with condition identity. : If present, the P-Served-User otherwise the P-Asserted-Identity value is matched with condition served-identity.	This is when AS chaining is enabled, OCB is checked in originating session case for served user B.  P-Served-User is used only if mtasSipSupportPServedUserHeader is set to 1.
12	re-INVITE	Transit	diverter	OCB	Request-URI: C P-Asserted-Identity: A	If present, the P-Served-User otherwise the P-Asserted-Identity value is matched with condition served-identity.	This is a re-INVITE in an existing call, no need to check outgoing barring. Barring also does nothing to a re-INVITE to C.  P-Served-User is used only if mtasSipSupportPServedUserHeader is set to 1.
13	re-INVITE	Transit Terminating	diverter	OCB	Request-URI: C P-Asserted-Identity: A	None	This is when AS chaining is disabled and there is a re-INVITE in an existing call, no need to check outgoing barring. Barring also does nothing to a re-INVITE towards A.
14	re-INVITE	Transit Originating	diverter	OCB	Request-URI: C P-Asserted-Identity: A P-Served-User: B	None	This is when AS chaining is enabled and there is a re-INVITE in an existing call, no need to check outgoing barring. Barring also does nothing to a re-INVITE towards A.





Table 2 BC Service Start

Id	Request	MTAS Type	Served User Role	Function	Header	Action	Comment
15	REFER	Transit	diverter	ICB	Request-URI: B  Referred-By: A  Refer-To: AS-A  P-Asserted-Identity: A	None	The diverter takes no part in the transfer, so no need to check incoming barring. Barring also does nothing to a REFER from C
16	REFER	Transit	diverter	OCB	Request-URI: C  Referred-By: A  Refer-To: AS-A  P-Asserted-Identity: A	None	The diverter takes no part in the transfer, so no need to check outgoing barring. Barring also does nothing to a REFER towards A.

There can be multiple P-Asserted-Identity headers.

One SIP request can include one or more identities, for example, SIP URIs, that are to be validated for CB, as shown in Table 2. This means that the service can be started multiple times by one request, for example, one per identity.

### 3.1 Anonymous Condition Checks

The algorithm used when checking for anonymity is shown in Figure 3.

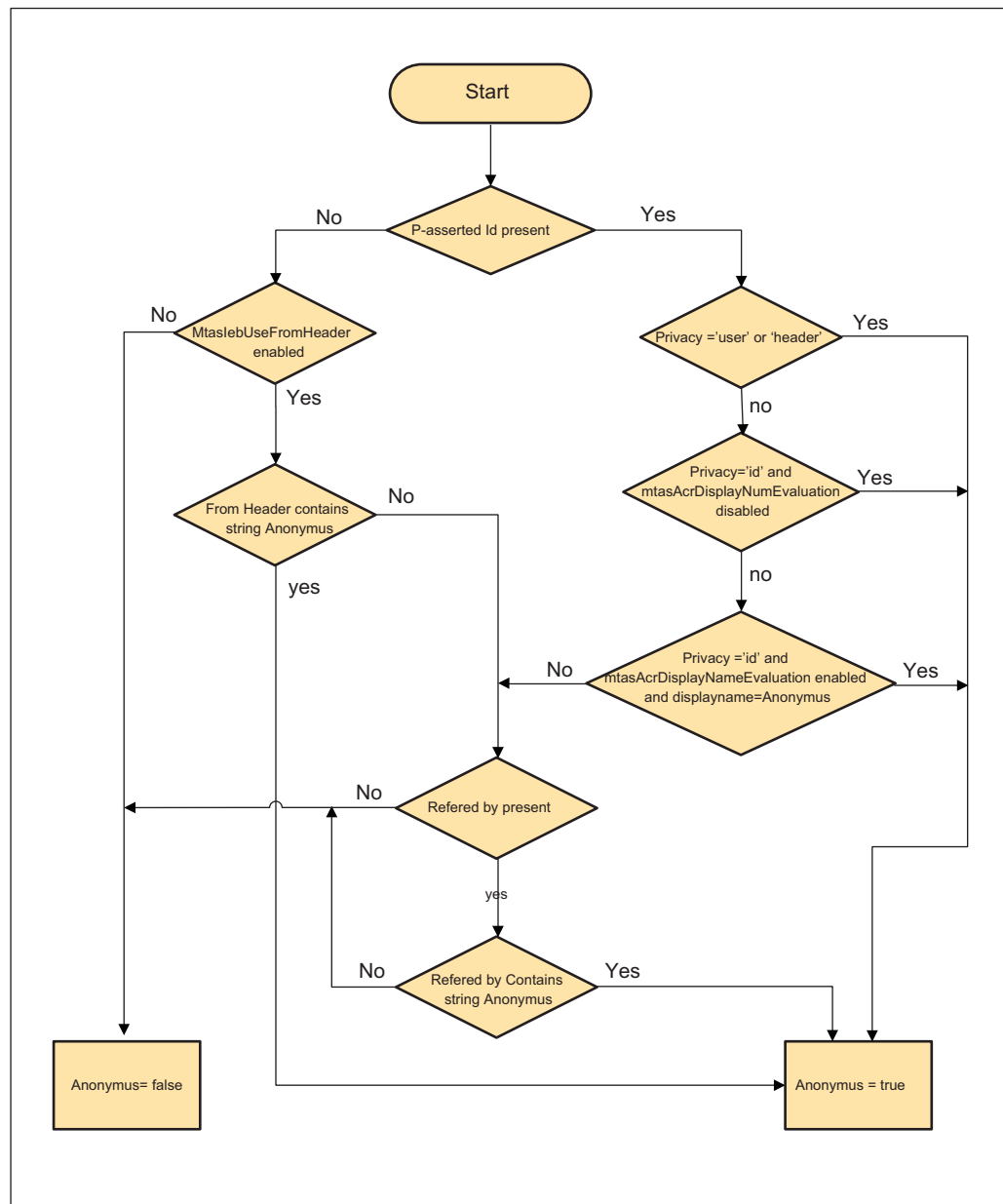


Figure 3 Anonymity Algorithm

## 3.2 Roaming Condition for Mobile User Checks

The following checks are done for the roaming condition for mobile users:

- The served user is a mobile user, that is, the user is provisioned with a Mobile Subscriber ISDN Number (MSISDN) on the HSS.
- The current country of the served user is different from its home country.



- The served user's current network is in the home country but belongs to different network.

For more information about Country Code Mapping, see [MTAS Country Code Mapping Management Guide](#).

### 3.3 International Condition for Mobile User Checks

The following checks are done for the international condition for mobile users:

- The served user is a mobile user, that is, the user is provisioned with an MSISDN on the HSS.
- The Req URI is a tel URI or an embedded tel in the global format.
- The country present in the Req URI is different from the current country of the served user.
- The country mapped from the country present in the Request URI is different from the served user's current country.

For more information about Country Code Mapping, see [MTAS Country Code Mapping Management Guide](#).

### 3.4 International-exHC Condition for Mobile User Checks

The following checks are done for the international-exHC condition for mobile users:

- The served user is a mobile user, that is, the user is provisioned with an MSISDN on the HSS.
- The Req URI is a tel URI or an embedded tel in the global format.
- The country present in the Req URI is different from both the home country of the served user and its current country.
- The country mapped from the country present in the Request URI is different from both the served user's home country and its current country.

For more information about Country Code Mapping, see [MTAS Country Code Mapping Management Guide](#).

### 3.5 Roaming, International, and International-exHC for Mobile User on Non-3GPP Access

- The served user is a mobile user that is determined by the `mtasMmtMobileUserDetermination` attribute or is provisioned with an MSISDN on the HSS.



- The home location of the served user is retrieved by `mtasFunctionMccMnc` configuration or by HSS MSISDN query.
- For mobile users on Non-3GPP access, the current location is considered to be the same as their home location and the Roaming, International and International-exHC barring rules are evaluated accordingly.

## 3.6 Checking for Call Analysis Condition

Call Analysis will bar the outgoing communications by evaluating the following conditions in operator barring rules:

- The B-Number classification field value in Number Translation rule to be assigned to the `b-number-type` element in the condition of barring rules.
- The B-Number classification field value in Number Translation rule to be assigned to `b-number-type` element in the condition of barring rules.
- The B-Number network type mapped by Number Portability service to be assigned to `b-network-type` element in the condition of barring rules.
- The Call type that can be determined by comparing the calling number and called number to the values in the `NumberAnalysis` MOM tree assigned to the `localness` element in the condition of barring rules.



## 4 Barring Rules

The MTAS offers several Barring services, which are controlled by the interactions of MOs and per subscriber data.

MOs and attributes mentioned in the following sections are further described in [Managed Object Model \(MOM\)](#).

Subscription barring rules are expressed in an XML document which consists of two parts, one for the served user and one for the operator. The operator barring rules override the served user barring rules. Service data is further described in [Section 5.19 Configure Service Data](#) on page 81.

### 4.1 Relationship between OCB Types and Dial Plans

The OCB and Dial Plan services control the destinations that a user is allowed to call. There are several types of Outgoing Communication Barring and dial plan. When a call attempt is evaluated by the different types of Outgoing Communication Barring, the evaluation stops when a result bar or allow is obtained. If no result is obtained, the call is allowed. The types of Outgoing Communication Barring are evaluated in the following order:

1. OTP Diversion Global blacklist

It allows the operator to bar calls being diverted to all addresses containing any of a list of strings.

This list is configured by setting the attribute `mtasCDivBlackList` in the `MtasCDiv` MO.

The whole of the normalized target Request URI is searched for each entry in `mtasCdivBlackList`. The normalized form of service numbers is local number (for example, `tel:151;phone-context=telco.com`). The normalized form of other numbers is global number (for example, `tel:+46107196992`). To disallow diversion of calls to a range of service numbers, for example, 110–119, include “`tel:11`” and “`sip:11`” in `mtasCdivBlackList`. To disallow diversion of calls to a range of other numbers, for example, all Swedish numbers, include “`+46`” in `mtasCdivBlackList`.

The use of wildcard character “`^`” (circumflex) is allowed in `mtasCdivBlackList` entries. Each occurrence of “`^`” matches any single character. For example, `:+46^^7196992` matches both `+46107196992` and `+46207196992`.

2. VTP Diversion Global blacklist

It allows the VTP operator to bar calls being diverted to all addresses containing any of a list of strings.



This list is configured by setting the attribute `vtasCdivBlackList` in the `VtasCdiv` MO.

The whole of the normalized target Request URI is searched for each entry in `vtasCdivBlackList`. The normalized form of service numbers is local number (for example, `tel:151;phone-context=telco.com`). The normalized form of other numbers is global number (for example, `tel:+46107196992`). To disallow diversion of calls to a range of service numbers, for example, 110–119, include “`tel:11`” and “`sip:11`” in `vtasCdivBlackList`. To disallow diversion of calls to a range of other numbers, for example, all Swedish numbers, include “`+46`” in `vtasCdivBlackList`.

The use of wildcard character “`^`” (circumflex) is allowed in `vtasCdivBlackList` entries. Each occurrence of “`^`” matches any single character. For example, `+46^^7196992` matches both `+46107196992` and `+46207196992`.

The attribute values in the `VtasCdiv` MO are only applied if the `vtasCdivDropBack` attribute is set to 0 (Use VTP values).

For more information about how to configure a VTP, see [MTAS Wholesale Support Management Guide](#).

### 3. OTP Global OCB White List

It allows the operator to allow calls to a set of addresses.

The set of addresses is configured by setting the attributes `mtasOcbWhiteListNumIncl`, `mtasOcbWhiteListNumExcl`, and `mtasOcbWhiteListDomainIncl` in the `MtasOcb` MO.

### 4. Nodal Dial Plan

It allows the operator to specify a set of addresses to which calls are permitted; calls to all other addresses are barred.

The set of addresses is configured by setting the attributes `mtasDialPlanAllowed`, `mtasDialPlanExcepted`, and `mtasDialPlanDomain` in the `MtasDialPlan` MO.

### 5. OTP Global OCB blacklist

It allows the operator to bar calls to all addresses containing any of a list of strings.

This list is configured by setting the attribute `mtasOcbBlackList` in the `MtasOcb` MO.

The whole of the normalized Request URI is searched for each entry in `mtasOcbBlackList`. The normalized form of service numbers is local number (for example, `tel:151;phone-context=telco.com`). The normalized form of other numbers is global number (for example, `tel:+46107196992`). To disallow calls to a range of service numbers, for example, 110–119, include



“tel:11” and “sip:11” in `mtas0cbBlackList`. To disallow calls to a range of other numbers, for example, all Swedish numbers, include “+46” in `mtas0cbBlackList`.

The use of wildcard character “^” (circumflex) is allowed in `mtas0cbBlackList` entries. Each occurrence of “^” matches any single character. For example, `:+46^^7196992` matches both `+46107196992` and `+46207196992`.

It is allowed, following a “[” character, to append a global announcement identity to the `mtas0cbBlackList` entry. It is a precondition that the announcement is configured as a `MtasGaAnn` generic announcement instance in the `MtasGa` MO. For example, if `toll_free_only_ann` is a configured `MtasGaAnn`, and `:+1^^^976^^^^|ann=toll_free_only_ann` is an OCB blacklist entry, then this configures that `toll_free_only_ann` is played when a call is barred because of its target matching `+1^^^976^^^^`.

For more information about how to configure a generic announcement, see [MTAS Generic Announcement Management Guide](#).

#### 6. OTP-controlled per-VTP Dial Plan

It allows the operator to specify a set of addresses to which users belonging to a VTP are permitted to call; calls to all other addresses are barred.

The set of addresses is configured by setting the attributes `mtasDpvAllowed`, `mtasDpvExcepted`, and `mtasDpvDomain` in the `MtasDpv` MO.

#### 7. VTP Global OCB White List

It allows the VTP operator to allow calls from a user belonging to that VTP, to a set of addresses.

The set of addresses is configured by setting the attributes `vtas0cbWhiteListNumIncl`, `vtas0cbWhiteListNumExcl`, and `vtas0cbWhiteListDomainIncl` in the `Vtas0cb` MO.

The values in the `Vtas0cb` MO are only used if the `vtas0cbDropBack` attribute is set to 0 (Use VTP values). This list is only evaluated if there exists a valid unexpired Wholesale license on the MTAS.

For more information about how to configure a VTP, see [MTAS Wholesale Support Management Guide](#).

#### 8. VTP-controlled per-VTP Dial Plan

It allows the virtual operator to specify a set of addresses to which calls are permitted; calls to all other addresses are barred.

The set of addresses is configured by setting the attributes `vtasDialPlanAllowed`, `vtasDialPlanExcepted`, and `vtasDialPlanDomain` in the `VtasDialPlan` MO.



For more information about how to configure a VTP, see [MTAS Wholesale Support Management Guide](#).

## 9. VTP Global OCB blacklist

It allows the VTP operator to bar calls from a user belonging to that VTP, to all addresses containing any of a list of strings.

This list is configured by setting the attribute `vtas0cbBlackList` in the `Vtas0cb` MO.

The whole of the normalized Request URI is searched for each entry in `vtas0cbBlackList`. The normalized form of service numbers is local number (for example, `tel:151;phone-context=telco.com`). The normalized form of other numbers is global number (for example, `tel:+46107196992`). To disallow calls to a range of service numbers, for example, 110–119, include “`tel:11`” and “`sip:11`” in `vtas0cbBlackList`. To disallow calls to a range of other numbers, for example, all Swedish numbers, include “`+46`” in `vtas0cbBlackList`.

The use of wildcard character “`^`” (circumflex) is allowed in `vtas0cbBlackList` entries. Each occurrence of “`^`” matches any single character. For example, `:+46^^7196992` matches both `+46107196992` and `+46207196992`.

It is allowed, following a “`|`” character, to append a global announcement identity to the `vtas0cbBlackList` entry. It is a precondition that the announcement is configured as a `MtasGaAnn` generic announcement instance in the `MtasGa` MO. For example, if `toll_free_only_ann` is a configured `MtasGaAnn`, and `:+1^^^976^^^^|ann=toll_free_only_ann` is an OCB blacklist entry, then this configures that `toll_free_only_ann` is played when a call is barred because of its target matching `+1^^^976^^^^`.

For more information about how to configure a generic announcement, see [MTAS Generic Announcement Management Guide](#).

The attribute values in the `Vtas0cb` MO are only used if the `vtas0cbDropBack` attribute is set to `0` (Use VTP values). This list is only evaluated if there exists a valid unexpired Wholesale license on the MTAS.

For more information about how to configure a VTP, see [MTAS Wholesale Support Management Guide](#).

## 10. Operator Barring Program or Operator Permitted Program

The Operator Barring Program is provisioned by the operator in the operator part of the Multimedia Telephony (MMTel) document of the subscriber. It allows the operator to define classes of telephone numbers and domains, and to bar calls to combinations of those classes of on a per subscriber basis.

The Operator Permitted Program is provisioned by the operator in the operator part of the subscribers Multimedia Telephony (MMTel) document.





It allows the operator to define groups of telephone numbers and domains, and to allow calls to combinations of those classes on a per subscriber basis. No further OCB checks are made if a call is NOT permitted by the Operator Permitted Program.

A subscriber cannot have both an Operator Barring Program and an Operator Permitted Program.

This program is configured by setting the attributes in the `Mtas0cb0pBCat` and `Mtas0cbBCat` MOs, and in the `NumberAnalysis` MO and its subordinate MOs.

#### 11. Operator Diversion Barring Program

The Operator Diversion Barring Program is provisioned by the operator in the operator part of the subscriber's MMTel document. It allows the operator to define classes of telephone numbers and domains, and on a per subscriber basis, bar calls being diverted to combinations of those classes of telephone numbers.

This program is configured by setting the attributes in the `Mtas0cb0pBCat`, `Mtas0cbBCat` MOs, and in the `NumberAnalysis` MO and its subordinate MOs.

#### 12. Operator outgoing barring rules for subscriber

These rules are provisioned by the operator in the operator part of the subscriber's MMTel document. It allows the operator to bar or allow calls by this subscriber based on combinations of the following criteria:

- Called identity
- Served identity
- Media type in request
- Date and time
- Carrier
- Roaming
- International
- International-exHC
- b-number-type
- b-network-type
- localness

The OCB rules of the caller connected on Wi-Fi or any Non-3GPP access type, are evaluated considering its current location to be the same as the home location.



No further OCB checks are made if a call is found to be explicitly allowed (Action: allow=true) by the operator outgoing barring rules for subscriber.

### 13. Barring Program

It is provisioned by the operator in the operator part of the subscriber's MMTel document, and provisioned by the subscriber in the user part of the subscriber's MMTel document. The Barring Program allows the operator to define groups of telephone numbers for which the subscriber is allowed to bar.

This program is configured by setting the attribute `mtas0cbNullBarringProgram` in the `Mtas0cb` MO, and by setting the attributes in the `Mtas0cbBCat`, `Mtas0cbSingleBp`, and `Mtas0cbMultipleBp` MOs, and in the `NumberAnalysis` MO and its subordinate MOs.

### 14. Subscriber outgoing barring rules

The Subscriber outgoing barring rules are provisioned by the subscriber in the user part of the subscriber's MMTel document, and allow the subscriber to bar or allow outgoing calls based on the following criteria:

- Called identity
- Served identity
- Media type in request
- Date and time
- Carrier
- Roaming
- International
- International-exHC

The following VTP-related OCB rules are only applied on users belonging to that VTP:

- VTP Diversion Global blacklist
- VTP Global OCB White List
- VTP-controlled per-VTP Dial Plan
- VTP Global OCB blacklist

If the served user is an OTP user, that is, it does not belong to any VTP, then only the OTP-related rules are performed on the communication. If the served user is a VTP user (the host part of its primary Public User Identity (PUI) belongs to a VTP domain), then both the OTP-related barring rules and the corresponding VTP-related rules are applied on the communication.



## 4.2 Relationship between ICB Types

The ICB service controls who can call a subscriber. There are several types of Incoming Communication Barring. When a call attempt is evaluated by the different types of Incoming Communication Barring, the evaluation stops when a result bar or allow is obtained. If no result is obtained, the call is allowed.

The types of ICB are evaluated in the following order:

### 1. Global ICB White List

It allows the operator to allow calls from a set of addresses.

The set of addresses is configured by setting the attributes `mtasIcbWhiteListNumIncl`, `mtasIcbWhiteListNumExcl`, and `mtasIcbWhiteListDomainIncl` in the `MtasIcb` MO.

### 2. Global Anonymous Communication Barring

It allows the operator to bar all incoming calls where the caller is anonymous.

This barring is configured by setting the attribute `mtasAcrGlobal` in the `MtasACR` MO.

### 3. Global ICB blacklist

It allows the operator to bar calls from all addresses that contain any part of a list of strings. An example of a list of string is as follows:

```
spam
otherbarredword
```

This list is configured by setting the attribute `mtasIcbBlackList` in the `MtasIcb` MO.

The whole of the URI part of each P-Asserted-Identity is searched for each entry in `mtasIcbBlackList`. The normalized form of service numbers is local number (For example, `tel:151;phone-context=telco.com`). The normalized form of other numbers is global number (for example, `tel:+46107196992`). To disallow calls from a range of service numbers, for example, 110–119, include “tel:11” and “sip:11” in `mtasIcbBlackList`. To disallow calls from a range of other numbers, for example, all Swedish numbers, include “+46” in `mtasIcbBlackList`.

### 4. VTP Global ICB White List

It allows the VTP operator to allow calls from a set of addresses. The list applies to all end users belonging to that VTP.

The set of addresses is configured by setting the attributes `vtasIcbWhiteListNumIncl`, `vtasIcbWhiteListNumExcl`, and `vtasIcbWhiteListDomainIncl` in the `VtasIcb` MO.

The attribute values in the `VtasIcb` MO are only used if the `vtasIcbDropBack` attribute is set to `0` (Use VTP values). This list is only evaluated if there exists a valid unexpired Wholesale license on the MTAS.

For more information about how to configure a VTP, see [MTAS Wholesale Support Management Guide](#).

## 5. VTP Global Anonymous Communication Barring

It allows the VTP operator to bar all incoming calls where the caller is anonymous. The list applies to all end users belonging to that VTP.

This barring is configured by setting the attribute `vtasAcrGlobal` in the `VtasACR` MO.

The attribute values in the `VtasACR` MO are only used if the `vtasAcrDropBack` attribute is set to `0` (Use VTP values).

For more information about how to configure a VTP, see [MTAS Wholesale Support Management Guide](#).

## 6. VTP Global ICB blacklist

It allows the VTP operator to bar calls from all addresses that contain any part of a list of strings. The list applies to all end users belonging to that VTP.

This list is configured by setting the attribute `vtasIcbBlackList` in the `VtasIcb` MO.

The whole of the URI part of each P-Asserted-Identity is searched for each entry in `vtasIcbBlackList`. The normalized form of service numbers is local number (For example, `tel:151;phone-context=telco.com`). The normalized form of other numbers is global number (for example, `tel:+46107196992`). To disallow calls from a range of service numbers, for example, 110–119, include “`tel:11`” and “`sip:11`” in `vtasIcbBlackList`. To disallow calls from a range of other numbers, for example, all Swedish numbers, include “`+46`” in `vtasIcbBlackList`.

The attribute values in the `VtasIcb` MO are only used if the `vtasIcbDropBack` attribute is set to `0` (Use VTP values). This list is only evaluated if there exists a valid unexpired Wholesale license on the MTAS.

For more information about how to configure a VTP, see [MTAS Wholesale Support Management Guide](#).

## 7. Operator incoming barring rules for subscriber

These rules are provisioned by the operator in the operator part of the subscriber’s MMTel document. It allows the operator to bar or allow calls to this subscriber based on combinations of the following criteria:

- Calling identity



- Served identity
- Media type in request
- Date and time
- Anonymity of caller
- Call has been diverted to this subscriber
- Roaming

Additionally, the operator can bar the incoming call for subscriber with Do-Not-Disturb action activated with any combination of these conditions.

No further ICB checks are made if a call is found to be explicitly allowed (Action: `allow=true`) by the operator incoming barring rules for subscriber.

#### 8. Subscriber incoming barring rules

These rules are provisioned by the subscriber in the user part of the subscriber's MMTel document. It allows the subscriber to bar (or allow) calls to this subscriber based on combinations of the following criteria:

- Calling identity
- Served identity
- Media type in request
- Date and time
- Anonymity of caller
- Call has been diverted to this subscriber
- Roaming

Also, the subscriber can bar the incoming call with Do-Not-Disturb action activated with any combination of these conditions.

When CLIR Interworking service is unlocked then if the calling party has Originating Identity Restriction active, then those subscriber defined ICB rules which use the identity of the caller are not evaluated.

For more information about Identity Presentation, see [MTAS Identity Presentation Management Guide](#).

The following VTP-related ICB rules are only applied on users belonging to that VTP:

- VTP Global ICB White List
- VTP Global Anonymous Communication Barring



#### — VTP Global ICB blacklist

If the served user is an OTP user, that is, it does not belong to any VTP, then only the OTP-related rules are performed on the communication. If the served user is a VTP user (the host part of its primary PUI belongs to a VTP domain), then both the OTP-related barring rules and the corresponding VTP-related rules are applied on the communication.

## 4.3 Global Barring Rules

The global barring rules are defined by MO attributes. There is one blacklist for global ICB, one blacklist for global OCB, and one blacklist for diverted global OCB. There is also an MO attribute to enable or disable the ACR on a global level. A list entry contains a string that is matched with the following (see also Table 2):

- P-Asserted-Identity in ICB
- Request URI in OCB
- Refer-To URI in OCB
- Referred-By URI in ICB
- From Header in ICB (if the CM attribute `mtasIcbUseFromHeader` is set to Enabled)

It is not required that the format of the entry is a SIP or tel URI. The matching of the list entry with a URI is true if and only if the list entry is a substring of the URI.

Example:

The following entries:

```
.se  
+468  
bob@example.com  
spam
```

match the following URIs:

```
sven@operator.se  
+468112233  
bob@example.com  
12345@good-spam.com
```

Matching is case-sensitive and US-ASCII is used as the character set.

If the ACR is enabled on the global level, all anonymous communications are rejected by the MTAS.



## 4.4 Global White List

The global white lists are defined by MO attributes. There is one white list for global OCB and one white list for global ICB. Both the Global OCB White List and Global ICB White List contain three lists of strings. (Number Included, Number Excluded, Domain Included).

- The “Number Included” list (`mtasOcbWhiteListNumIncl` and `mtasIcbWhiteListNumIncl`) specifies the leftmost parts of the normalized numbers that are included in the global white list. This list is only compared with a tel URI or a SIP URI containing a telephone number. Each entry in the list contains a string which represents the left-most part of a number. A string representing a global number must be prefixed by a “+”. A string representing a local number can include the phone-context parameter.
- The “Number Excluded” list (`mtasOcbWhiteListNumExcl` and `mtasIcbWhiteListNumExcl`) specifies the leftmost parts of the normalized numbers that are excluded from the global white list. This list is only compared with a tel URI or a SIP URI containing a telephone number. Each entry in the list contains a string which represents the left-most part of a number. A string representing a global number must be prefixed by a “+”. A string containing a local number can include the phone-context parameter. Each string must begin with one of the strings in the ‘Number included’ list (but this is not policed).
- The “Domain Included” list (`mtasOcbWhiteListDomainIncl` and `mtasIcbWhiteListDomainIncl`) specifies the set of domains that is included in the global white List. This list is only compared with a SIP URI that does not contain a telephone number. Each entry in the list contains a string which represents the host part of a URI. If the first character in the string is a “\*”, this is to be treated as a wildcard character and a rightmost match of the domain name from the remote identity is performed with the rest of the characters in the string. If the first character in the string is not a “\*”, then the domain name from the remote identity must exactly match the included string.

The Global ICB White list and the Global OCB White list are subject to the MMTel Extended License and are only evaluated if there is a valid unexpired MMTel Extended License available on the MTAS node.

## 4.5 Dial Plan

The Nodal Dial Plan, the OTP-controlled per-VTP Dial Plan, and the VTP-controlled per-VTP Dial Plan are defined as MO attributes. Each dial plan is defined by a list of allowed numbers and a list of excepted numbers, which apply to numerical Request URIs (that is, SIP URIs with a user=phone parameter and tel URIs), and a list of allowed domains, which applies to non-numerical Request URIs.



## 4.6 Location Based Barring

The Location Based Barring is configured on node level by MO attributes and it defines a list of number-prefixes that are to be barred from a specific location area and a list of prefixes which are exempted from barring. It also specifies the announcement to be played when a call is barred.

The `Mtas0cbLb` MO represents the Location Based OCB (OCBLB) service in an MTAS node when the `Vtas0cbLb` MO represents the OCBLB service for a VTP MTAS node.

The instance of an `mtas0cbLb` MO class consists of:

- List of B-Number prefixes to be barred – defines the leftmost parts of the global numbers and local numbers whose calling is barred from a specific location area. It is a list of strings where each string represents one leftmost part. The B-Number prefixes can contain digits and + sign only.
- List of exemptions for B-Number prefixes – This attribute defines the leftmost parts of the global numbers and local numbers which are exempted from barring. It is a list of strings where each string represents one leftmost part. Each string begins with one of the strings in the list of B-Number prefixes to be barred. The attribute cannot include numbers that are not already covered by the list. The B-Number prefixes can contain digits and + sign only.
- Announcement name – defines the name of the generic announcement to be used when a communication attempt is rejected by Location Based OCB. If this attribute is empty or does not specify an instance of `MtasGaAnn`, no announcement is played.
- Announcement control – defines whether the value specified in announcement name is used to determine the announcement to play when the Location Based OCB rejects a communication attempt.

When the MTAS receives an INVITE with PANI header, it checks if the called number is barred because of Location Based OCB.

## 4.7 Relationship between Barring Programs and User Barring Categories

A Barring Category defines a group of telephone numbers to be barred. A Barring Program is a combination of Barring Categories. There are two Barring Program schemes; “Single”, where a subscriber can select one of the Barring Programs defined by the operator, and “Multiple”, where a subscriber can select any combination of the Barring Programs defined by the operator.

Up to 16 User Barring Categories can be defined by setting data in the `Mtas0cbBCat` MO. There are three Localness Barring Categories named “Local”, “Non Local”, and “Allow Local”, which are defined by setting data in the `NumberAnalysis` MO and its subordinate MOs. The User Barring Categories and





Localness Barring Categories are used by Barring Programs in both the Single and the Multiple Barring Program schemes. For more information, see [Managed Object Model \(MOM\)](#).

For the Single barring programs scheme, up to 256 Barring Programs can be defined. Barring Categories are combined into Barring Programs by the operator through the MOC `mtas0cbSingleBp`, and the MOs `mtas0cbSingleBpCategories`, `mtas0cbSingleBpLocalBCats`, and `mtas0cbSingleBpZone1Cats`. It is recommended to assign a Barring Category to only one Barring Program to avoid activation and deactivation of multiple Barring Categories with one SSC command.

For the Multiple barring programs scheme, up to 16 Barring Programs can be defined. Barring Categories are combined into Barring Programs by the operator through the MOC `mtas0cbMultipleBp`, and the MOs `mtas0cbMultipleBpCategories`, `mtas0cbMultipleBpLocalCats`, and `mtas0cbMultipleBpZone1Cats`. It is recommended to assign a Barring Category to only one Barring Program to avoid activation and deactivation of multiple Barring Categories with one SSC command.

The Barring Program scheme applicable to a subscriber is defined in the operator part of the subscriber's MMTel document.

For a subscriber allocated to the Single Barring Program scheme, the Barring Program currently applicable is defined in the user part of the subscriber's MMTel document.

For a subscriber allocated to the Multiple Barring Program scheme, the set of User and Localness Barring Categories currently applicable is defined in the user part of the subscriber's MMTel document.

Handling of the Localness Barring Categories "Local", "Non Local" and "Allow Local":

- If the set of categories contains "Local", "Non Local", and "Allow Local", the "Allow Local" category is ignored and the call is barred.
- If the set of categories contains "Local" and "Allow Local", the "Allow Local" category is ignored. If the call is determined to be local, the call is barred.
- If the set of categories contains "Local" and "Non Local", the call is barred.
- If the set of categories contains "Non Local" and "Allow Local", and the call is determined to be local, the call is not barred by this level of barring. Otherwise, the call is barred.
- If the set of categories contains any one of the special Barring Categories "Local", "Non Local", and "Allow Local", the checking is performed as the following description:



Whether a call is local, is determined through the values in the `NumberAnalysis` MO and its subordinate MOs, which are described in [Managed Object Model \(MOM\)](#).

To check whether a call is allowed by the special Barring Category “Allow Local”, if the call is determined to be local, the call is not barred by Barring Programs. Otherwise, the checking moves on to the next Barring Category.

To check whether a call is barred by the special Barring Category “Local”, if the call is determined to be local, the call is barred. Otherwise, the checking moves on to the next Barring Category.

To check whether a call is barred by the special Barring Category “Non Local”, if the call is determined to be non-local, the call is barred. Otherwise, the checking moves on to the next Barring Category.

To check whether a call is barred by the Zone1 Barring Categories, the call type determined with the `NumberAnalysis` MOM is checked as follows:

- “L\_National”, bar if Number Analysis call type is `LongDistance`
- “L\_International”, bar if Number Analysis call type is `International`
- “L\_IntraLata”, bar if Number Analysis subcall type is `IntraLata`
- “L\_IntraLataToll”, bar if Number Analysis subcall type is `IntraLataToll`
- “L\_InterLata”, bar if Number Analysis call type is `LongDistance`
- “L\_NanpZone1”, bar if Number Analysis subcall type is `NanpZone1`
- “L\_Nanp”, bar if Number Analysis call type is `NanpInternational`

In each case, the announcement indicated by the corresponding instance of `mtasOcbLocalnessBCatAnnouncementName` and `mtasOcbLocalnessBCatAnnouncementControl` is played.

If the called party number is on the list of exempted numbers, `MtasNumNormLocalnessExemptListNumber`, then the call is allowed and barring is bypassed. For more details, see [MTAS Number Normalization Management Guide](#). If the call type is already determined by Dialed Number Mapping, then Communication Barring service uses that call type, and does not try to determine the call type.

Localness barring is bypassed if the number was not normalized to a global number starting with +.

#### 4.7.1 Subscriber Data

The use of the Barring Programs feature is defined in XML, and is part of the transparent data of the user, held on the HSS. Outgoing Barring Programs has one subscriber part and one operator part, as illustrated in Figure 4 and Figure 5.

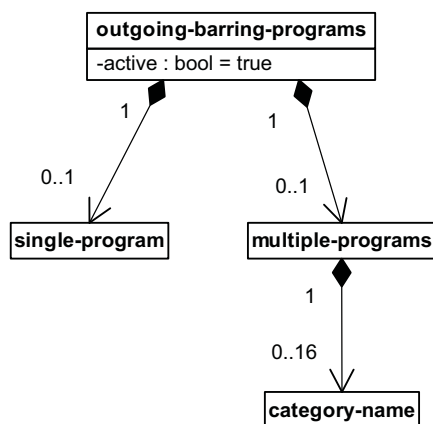


Figure 4 Subscriber Outgoing Barring Programs Elements

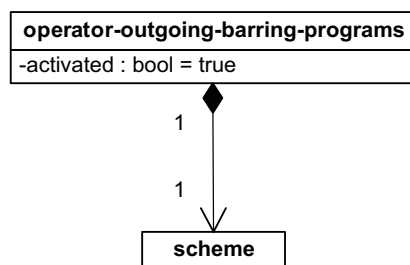


Figure 5 Operator Outgoing Barring Programs Elements

A subscriber outgoing barring programs element contains either a single program element or a multiple programs element; which is allowed depends on the value of the scheme element in the operator Barring Programs elements.

## 4.7.2 Supplementary Service Codes

The activation, deactivation, and interrogation of Barring Programs through the SSCs are described in [MTAS Supplementary Service Codes Management Guide](#).

### Single Scheme

Activation	The Barring Program number entered in the SSC is copied to the single program element of the subscriber outgoing barring programs element.
Deactivation	The Barring Program number in the MO <code>mtas0cbNullBarringProgram</code> is copied to the single program element of the subscriber outgoing barring programs element.

### Multiple Scheme



## Activation

The Categories which make up the Barring Program identified by the number in the SSC are determined by reading the MOC Mtas0cbMultipleBp, which gives a list of Barring Category numbers, a list of Zone1 Categories, and an indication of whether the categories Local, Non Local, and Allow local are part of the Program. For each entry in the list, the name of the Category is retrieved from MO mtas0cbBCatName. If the name does not already appear in the Category Name elements of the subscriber outgoing barring programs element, a new Category Name element is created containing the name of the Category.

For each entry in mtas0cbMultipleBpZone1Cats, if the name does not already appear in the Category Name elements of the subscriber outgoing barring programs element, a new Category Name element is created containing the value of the entry.

- If the MO mtas0cbMultipleBpLocalCats is 1 or 3, a new Category Name element containing “Local” is created in the outgoing-barring-programs element, if one does not exist.
- If the MO mtas0cbMultipleBpLocalCats is 2 or 3, a new Category Name element containing “Non Local” is created in the outgoing-barring-programs element, if one does not exist.
- If the MO mtas0cbMultipleBpLocalCats is 4, a new Category Name element containing “Allow Local” is created in the outgoing-barring-programs element, if one does not exist.

## Deactivation

The Categories which make up the Barring Program identified by the number in the SSC are determined by reading the MOC Mtas0cbMultipleBp, which gives a list of Barring Category numbers, a list of Zone1 Categories and an indication of whether the categories Local, Non Local, and Allow local are part of the Program. For each entry in the list, the name of the Category is retrieved from MO mtas0cbBCatName. If the name is displayed in the Category Name elements of the subscriber outgoing barring programs element, that Category Name element is deleted.

For each entry in mtas0cbMultipleBpZone1Cats, if the name is displayed in the Category Name elements of the subscriber outgoing barring programs element, that Category Name element is deleted.

- If the MO mtas0cbMultipleBpLocalCats is 1 or 3 and the outgoing-barring-programs element contains a Category Name element containing “Local”, that Category Name element is deleted.
- If the MO mtas0cbMultipleBpLocalCats is 2 or 3 and the outgoing-barring-programs element contains a Category Name element containing “Non Local”, that Category Name element is deleted.
- If the MO mtas0cbMultipleBpLocalCats is 4 and the outgoing-barring-program element contains a Category Name element containing “Allow Local”, that Category Name element is deleted.



## 4.8 Relationship between Operator Barring Programs, Operator Permitted Programs, and Operator Diversion Barring Programs

Operator Barring Programs, Operator Permitted Programs and Operator Diversion Barring Programs are based on defining a set of Operator Barring Categories, which supplement the Barring Categories defined for user-accessible Barring Programs. An Operator Barring Category is defined by a list of numbers and domains to be matched, defined by the MOs `mtas0cb0pBCatNumBarred` and `mtas0cb0pBCatDomain`, and a list of numbers to be exempted from the barring, defined by the MO `mtas0cb0pBCatNumExempted`. There are 10 special Barring Categories named “Local”, “Non Local”, “L\_National”, “L\_International”, “L\_IntraLata”, “L\_IntraLataToll”, “L\_InterLata”, “L\_NanpZone1”, “L\_Nanp”, and “Allow Local”, which are defined by setting data in the MO `NumberAnalysis`, see *Managed Object Model (MOM)*.

A user’s Operator Barring Program is defined in the operator part of the XML file of the user. The element contains a list of category names; the list can consist of Operator Barring Category Names, user Barring Category Names, and special Barring Category names. This allows the operator to reuse user Barring Categories in the Operator Barring Program, and to have Barring Categories that are not accessible to the user.

A user’s Operator Permitted Program is defined in the operator part of the user’s XML file instead of an Operator Barring Program. The element contains a list of category names; the list can consist of Operator Barring Category Names, user Barring Category Names, and special Barring Category Names except “Allow Local”.

When the user attempts to make an outgoing call, the number is extracted from the Request URI and checked against the set of Operator Barring Categories, Barring Categories and special Barring Categories specified for the user. If the Request URI does not contain a telephone number, then the domain is extracted and checked against the set of Operator Barring Categories.

A user’s Operator Diversion Barring Program is defined in the operator part of the user’s XML file. The element contains a list of category names; the list can consist of, Operator Barring Category Names, user Barring Category Names, and special Barring Category Names.

When the user attempts to divert an incoming call, the number is extracted from the Request URI and checked against the set of Operator Barring Categories, user Barring Categories, and special Barring Categories specified for the user in the union of the Operator Barring Program and the Operator Diversion Barring Program. If the Request URI does not contain a telephone number, then the domain is extracted and checked against the set of Operator Barring Categories.

**Note:** Which special Barring Categories a call is in is determined with the values in the `NumberAnalysis` MO and its subordinate MOs, which are described in *Managed Object Model (MOM)*.



If the set of categories contains any of “L\_National”, “L\_International”, “L\_IntraLata”, “L\_IntraLataToll”, “L\_InterLata”, “L\_NanpZone1”, “L\_Nanp”, the call type is determined by reference to the MOC NumberAnalysis and its subordinates. If the call type matches the category, the call is barred, and the announcement indicated by the corresponding instance of `mtasOcbLocalnessBCatAnnouncementName` and `mtasOcbLocalnessBCatAnnouncementControl` is played.

To check if a number or domain is barred by an Operator Barring Program or permitted by an Operator Permitted Program, the number is checked against each Operator Barring Category and Barring Category or the domain is checked against each Operator Barring Category until a match is found, or all Operator Barring Categories and Barring Categories in the set have been checked.

To check if a number is a match for an Operator Barring Category or Barring Category, the number is prefix matched against each entry in the list of numbers to be barred. If a match is found, the number is prefix matched or fixed length matched against each entry in the list of numbers to be exempted. An entry can include one or more wildcard characters (^) at any position in the number interpreted as character values. An entry can include the \$ character at the last position for fixed length number matching. A number is barred if a match is found in the barred list and no match is found in the exempted list. If a match is found in the exempted list, the number is not barred, and the checking continues with the next Operator Barring Category or Barring Category.

To check if a domain is a match for an Operator Barring Category, the host part of the URI is whole string matched against the list of allowed domains, each of which can contain a single wildcard at the beginning.

#### 4.8.1

#### Subscriber Data

An operator’s use of the Operator Barring Programs feature is defined in XML, and is part of the user’s transparent data held on the HSS. The Operator Barring Program has one operator part, as shown in Figure 6.

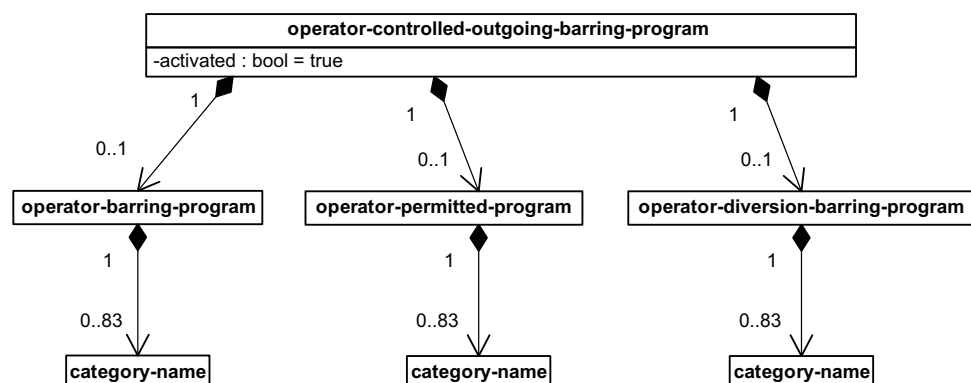


Figure 6 Operator Outgoing Barring Programs Elements

## 4.9 Subscription Rules

The subscription rules, defined by XML, are managed per ICB and OCB. ACR and DNDCB are a kind of ICB and its rules are defined under ICB.

The XML is defined by a schema and follows the structure illustrated in Figure 7 and Figure 8.

ICB and OCB have one subscriber rule set each, and one operator rule set each.

Subscription rules are subject to the MMTel Extended license and are only evaluated if there is a valid unexpired MMTel Extended license available on the MTAS node.

The rule set consists of zero or more rules, and each rule consists of zero or more conditions and one or more actions.

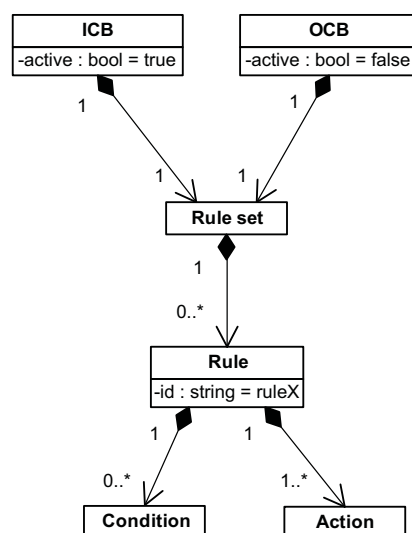


Figure 7 Subscriber Barring Rules Elements

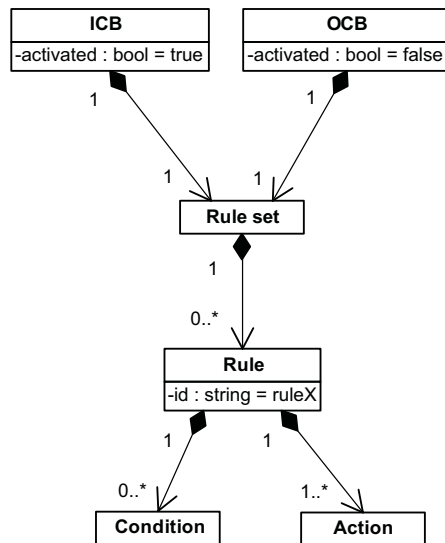


Figure 8 Operator Barring Rules Elements

The rule has one mandatory action named `allow` of type Boolean. The following applies for the values:

- `true` – allow the communication
- `false` – bar the communication

The other actions that apply to barring is:

- `do-not-disturb`

This action is used to indicate that the rule is a DNDCB rule. This action is applicable only for ICB rules. The `do-not-disturb` action cannot be combined with `allow=true` action within the same rule.

- `play-announcement`

This action is used to indicate Communication Barring service to play announcement by using Generic Announcement service in case of successful barring. The `play-announcement` action contains the selected operator named string value to be played to the calling party. The `play-announcement` action cannot be combined with `allow=true` action within the same rule.

- `play-segmented-announcement`

This action is used to indicate Communication Barring service to play a segmented announcement by using Generic Announcement service in case of successful barring. The `play-segmented-announcement` action contains the selected operator named string value to be played to the calling party and the optional list of announcement voice variable name/value pairs. The `play-segmented-announcement` action cannot be combined with `allow=true` action within the same rule.





The element Rule has one attribute id that identifies the rule. It has nothing to do with the evaluation order. The top-level ICB/OCB elements each has an attribute active of type Boolean. If active=true, then the rule set is evaluated if the corresponding service invocation is executed, and if active=false, then the rule set is not evaluated.

If the settings in Figure 7 are applied, then, for example, service invocation 1 in Table 2 does not evaluate any rules (since active=false for OCB), but service invocation 6 in Table 2 evaluates rules (if the ICB rule set has any rules defined). If the active attribute is not present, its default value is true.

The conditions that apply to barring are as follows:

— cp:identity

For ICB/ACR/DNDCB, this condition evaluates to true when the calling user's identity matches with the value of the identity element. For OCB, this condition evaluates to true when the called user's identity matches the value of the identity element. In all other cases, the condition evaluates to false. The interpretation of the special cases of a <one id> element containing a hidden: URI and a <one id> element containing a <number-match> element are described later in the text. The interpretation of all the other elements of this condition is described in [IETF RFC 4745](#).

If there is more than one URI, then CB iterates over all the URIs and evaluates the identity condition, if any URI matches then the condition evaluates to true. The URIs used to match this condition are described in Table 2.

The comparison of tel URIs is based on Section 4 in [IETF RFC 3966](#).

However, for CB, the input tel URI is only considered equal to the <one id> or <except> element in the cp:identity condition (the rule URI) if the number parts match and the parameters satisfy the following conditions:

- If there is a phone context=<pvalue> parameter in the rule URI, then there must be an identical phone context=<pvalue> parameter in the input URI.
- If there is no phone-context parameter in the rule URI, then there must be no phone context parameter in the input URI.
- If there is an ext=<phonedigits> parameter in the rule URI, then there must be an identical ext=<phonedigits> parameter in the input URI.
- If there is no ext parameter in the rule URI, then any ext parameter in the input URI is ignored.
- If there is an isub=<subaddress> parameter in the rule URI, then there must be an identical isub=<subaddress> parameter in the input URI.
- If there is no isub parameter in the rule URI, then any isub parameter in the input URI is ignored.



- Any other parameters are ignored.

For CB, the input tel URI is considered to match the `<number match>` element in the `cp:identity` condition (the rule URI) if the first few characters of the number part of the input URI match all the characters in the rule URI. Parameters cannot appear in the `<number match>` element, so any parameter in the input URI is ignored.

The comparison of SIP URIs is based on section 19.1.4 in [IETF RFC 3261](#).

However, only the user and host parts of SIP URIs are considered when comparing for CB, that is, the password, port, uri parameters, and headers are ignored for comparing SIP URIs for CB. For a SIP URI that was converted from a tel URI, the user part of the SIP URI contains the number and parameters of a tel URI. The comparison must first consider the host part of the SIP URI, and then must treat user part as if it were a tel URI.

The comparison of hidden URIs is performed as if there were a `<one id>` element in the identity condition for each identity in the corresponding entry in the `<operator-dynamic-black-list>` element.

When CLIR Interworking service is unlocked and if the calling party has Originating Identity Restriction active, then those subscriber defined ICB rules which use the identity of the caller are not evaluated.

For more information about Identity Presentation, see [MTAS Identity Presentation Management Guide](#).

#### — `mmt-serv:served-identity`

This condition evaluates to true when one of its `mmt-serv:one` subelements match the served user's identity and a valid Multi-Subscriber Number license is available. The `mmt-serv:one` subelements are interpreted similar to those of `cp:identity`.

The served user is determined in accordance with Section 5.7.1.3A of the [3GPP TS 24.229 v11.6.0](#).

- The P-Served-User header field is to be used, if present.
- Otherwise, the content of the Request URI (on the terminating side) or the P-Asserted-Identity header field (on the originating side) is to be used.

If the served user cannot be determined, service rules using the served-identity conditions are ignored:

- For each `served-identity` condition, it is checked if the user has any matching alias in IRS. This is done by iterating through all PUIs in IRS and matching them against the `served-identity` condition. There must be at least one PUI which matches.
- Within a `served-identity` condition of a Communication Diversion rule, each one element must have a unique id value.



- The served-identity element can contain only the `mmt-serv:one` subelements.
- `anonymous`

For ICB, this condition is evaluated as shown in Figure 3. This condition is not allowed in rules in OCB.
  - `roaming`

For both OCB and ICB, this condition evaluates to `true`, see Section 3.2 Roaming Condition for Mobile User Checks on page 20.
  - `international`

For OCB, this condition evaluates to `true`, see Section 3.3 International Condition for Mobile User Checks on page 21. This condition is not allowed in rules in ICB.
  - `international-exHC`

For OCB, this condition evaluates to `true`, see Section 3.4 International-exHC Condition for Mobile User Checks on page 21. This condition is not allowed in rules in ICB.
  - `cp:validity`

Specifies one or more periods. The condition evaluates to `true` when the current time is within the validity period expressed by one of the time periods included in this condition. In all other cases, the condition evaluates to `false`.

It expresses the rule validity period by two attributes, `<from>` containing a starting time and `<until>` containing an ending time. The validity condition is `true` if the current time is greater than or equal to at least one `<from>` child, and less than the `<until>` child after it.

The format is XML `dateTime`. Its canonical representation is UTC and time zones are expressed as a positive or negative duration. `2005-12-24T12:00` in Stockholm, GMT+1, is expressed as `2005-12-24T12:00:00+01:00` and has the corresponding canonical representation `2005-12-24T11:00:00Z`.

When the validity period is given in local time format, the UTC offset is taken from the `user-common-data`. If the UTC offset is not provisioned for the user, the value from the CM attribute `mtasMmtCalUtcOffset` is used.

When setting the value of these attributes the Daylight Saving Time correction must also be considered.

The recommended way of specifying time is to use local time format in the condition and use the UTC offset from the CM attribute `mtasMmtCalUtcOffset`.

For more information about MMTel, see [MTAS MMTel Management Guide](#).



Times in `cp:validity` conditions are converted to UTC before being compared to the current time, also in UTC.

— `mmt-serv:invalidity`

Specifies one or more periods. The condition evaluates to `false` when the current time is within the validity period expressed by one of the time periods included in this condition. In all other cases, the condition evaluates to `true`.

It expresses the rule invalidity period by two attributes, a starting and an ending time. The invalidity condition is `false` if the current time is greater than or equal to at least one `<from>` child, and less than the `<until>` child after it. The invalidity condition is `true` only when the current time is not within any of the invalidity periods.

The format is XML `dateTime`. Its canonical representation is UTC and time zones are expressed as a positive or negative duration. (2005-12-24T12:00:00 in Stockholm, GMT+1, is expressed as 2005-12-24T12:00:00+01:00 and has the corresponding canonical representation 2005-12-24T11:00:00Z.) When the invalidity period is given in local time format, the UTC offset is taken from the user-common-data. If the UTC offset is not provisioned for the user, the value from the CM attribute `mtasMmtCalUtcOffset`, is used.

When setting the value of these attributes the Daylight Saving Time correction must also be considered.

The recommended way of specifying time is to use local time format in the condition and use the UTC offset from the CM attribute `mtasMmtCalUtcOffset`.

Times in `mmt-serv:invalidity` conditions are converted to UTC before being compared to the current time, also in UTC.

For more information about XML `dateTime`, see [XML Schema Part 2: Datatypes Second Edition](#).

For more information about MMTel, see [MTAS MMTel Management Guide](#).

— `mmt-serv:valid-periods`

Allows assembly of complex time conditions based on one of the following subconditions:

- Times of day (in form of intervals, like 09:00-12:00)
- Days of the week
- Workdays/non-workdays (use of pre-provisioned/configured list of weekdays)
- Private and public holidays (use of pre-provisioned/configured list of holidays)



- Days of the month (for example, first day, last day, first Wednesday, the 13th)
- Calendar months
- Calendar weeks
- Daily, weekly, and monthly repetitions (with start day and repetition interval)

If any of the elements within the subconditions evaluate to `true`, then the subcondition evaluates to `true`.

The `mmt-serv:valid-periods` evaluates to `true` only when all the included subconditions evaluates to `true`.

It is also possible to mark the holidays as exception for the whole `mmt-serv:valid-periods` condition. So, when the current day is holiday then the `mmt-serv:valid-periods` evaluates to `false`.

The `mmt-serv:valid-periods` allows definition of UTC offset that is used for converting the time periods within the condition to UTC. If the UTC offset is not included in the condition, the UTC offset is taken from the `user-common-data`. If the UTC offset is not provisioned for the user, the value from the CM attribute `mtasMmtCalUtcOffset`, is used.

When setting the value of these attributes the Daylight Saving Time correction must also be considered.

The recommended way of specifying time is to skip the UTC offset in the condition and use the UTC offset from the CM attribute `mtasMmtCalUtcOffset`.

For more information about MMTel, see [MTAS MMTel Management Guide](#).

#### — media

This condition evaluates to `true` when the value of this condition matches the media field in one of the `m=` lines offered in an INVITE request, see [IETF RFC 4566](#).

It allows for barring of specific media.

#### — communication-diverted

This condition evaluates to `true` when the incoming communication has been previously diverted. This condition is not allowed in rules in OCB.

**Note:** Diverted communication can be recognized by the presence of the History-Info header field. For more information, see [MTAS Communication Diversion Management Guide](#).

#### — rule-deactivated



This condition always evaluates to false. This can be used to deactivate a rule, without losing information. By removing this condition the rule can be activated again.

— `ocp:other-identity`

If present in any rule, the `other-identity` element, which is empty, match all identities that are not referenced in any rule. It allows for specifying a default policy.

— `mmt-serv:carrier`

This condition consists of the following two optional elements:

- The `<carrier-select-code>` element contains the dialed Carrier Select Code. The operator can allow/disallow the use of the `<carrier-select-code>` element for the user and a complete match is done.
- The `<carrier-name>` element contains an alias name of the carrier selected for the call on call-by-call basis.

The `mmt-serv:carrier` condition is not allowed in rules in ICB. For OCB, this condition evaluates to true if the carrier selected by the Carrier Select Rn (CSRn) service is matching to the `carrier-name` or `carrier-selection-code` in the condition.

— `mmt-serv:subscriber-state`

The `mmt-serv:subscriber-state` condition is evaluated to true when the operator-configured `mmt-serv:subscriber-state` in the barring rule is matched with the operator-provisioned subscriber-state in the `mmt-op:subscriber-state`. The `mmt-op:subscriber-state` is provisioned in the `mmt-op:operator-common-data`. This condition is applicable for both ICB and OCB users.

— `mmt-serv:b-number-type`

This condition evaluates to true when the operator-configured `mmt-serv:b-number-type` in the barring rule is matched with B-Number classification field returned by Number Translation service of MMTel AS. This condition is applicable for OCB user only.

— `mmt-serv:b-network-type`

This condition evaluates to true when the operator-configured `mmt-serv:b-network-type` in the barring rule is matched with network type mapped for returned RN from Number portability query result by Number Portability Service. This condition is applicable for OCB user only.

— `mmt-serv:localness`



This condition evaluates to true when the operator-configured `mmt-serv:localness` in the barring rule is matched with localness mapped from call types determined by comparing the calling number and called number to the values in the NumberAnalysis MOM tree. This condition is applicable for OCB user only.

## 4.9.1 Supplementary Service Codes

The invocation, deactivation, and interrogation of Dynamic Black List and Malicious Communication Rejection through the SSCs are described in MTAS Supplementary Service Codes Management Guide.

### Dynamic Black List

Invocation	<p>The invocation command contains an optional parameter indicating whether the last caller, or the second last caller, is to be added to the Dynamic Black List. If the parameter is omitted, the last caller is added to the Dynamic Black List.</p> <p>If the caller's identity was available to the served user, a new rule is added to the served user's subscriber incoming barring rules with a single condition identity containing all the caller's identities, and with <code>allow = false</code>.</p> <p>If the caller's identity was not available to the served user, a new rule is added to the served user's subscriber incoming barring rules with a single condition identity containing a hidden: URI, and with <code>allow = false</code>, and an entry is added to the Dynamic Black List element in the operator part of the user's MMTel document, containing all the caller's identities. The new entry in the Dynamic Black List is identified by the date and time of creation in UTC. The value of the hidden: URI is a string representation of the date and time of creation of the Dynamic Black List entry.</p>
Deactivation	<p>All the entries in the Dynamic Black List created by invocation of Dynamic Black List are deleted. All the rules in the served user's subscriber incoming barring rules created by invocation of Dynamic Black List (both those containing the caller's identities and those containing a hidden: URI) are deleted.</p>

### Malicious Communication Rejection

The Malicious Communication Rejection invocation SSC allows the user to combine the actions of MCID and Dynamic Black List in a single SSC. For details of MCID, see MTAS Malicious Communication Identification Management Guide.



Invocation	<p>The invocation command contains an optional parameter indicating whether the last caller, or the second last caller, is to be subject to Malicious Communication Rejection. If the parameter is omitted, the last caller is the subject.</p> <p>The caller is reported as malicious.</p> <p>If the caller's identity was available to the served user, a new rule is added to the served user's subscriber incoming barring rules with a single condition identity containing all the caller's identities, and with <code>allow = false</code>.</p> <p>If the caller's identity was not available to the served user, a new rule is added to the served user's subscriber incoming barring rules with a single condition identity containing a hidden: URI, and with <code>allow = false</code>, and an entry is added to the Dynamic Black List element in the operator part of the user's MMTel document, containing all the caller's identities. The new entry in the Dynamic Black List is identified by the date and time of creation in UTC. The value of the hidden: URI is a string representation of the date and time of creation of the Dynamic Black List entry.</p>
Deactivation	<p>All the entries in the Dynamic Black List created by invocation of Malicious Communication Rejection are deleted. All the rules in the served user's subscriber incoming barring rules created by invocation of Malicious Communication Rejection (both those containing the caller's identities and those containing a hidden: URI) are deleted.</p>

## 4.10 Evaluation of Rules

If the rule set is active, then the CB evaluates the rules. The rules are evaluated from top to bottom, for example, rule *n* precedes rule *n + 1*.

All the rules in the rule set are evaluated to test if their respective conditions are true. A rule is said to be matched if all conditions evaluate to `true`. This means that the evaluation of conditions stops when the first condition is false (lazy evaluation).

If exactly one rule matches, then its specified action is executed, for example, if `allow` is as follows:

- `true` – the call continues normally
- `false` – the call is barred

The action part is only performed when the rule is matched. For example, the following rule:

```
validity = 10.00-11.00 2005-12-14
```





`allow = false`

only means that the communication is barred between 10.00-11.00 2005-12-14. It does not mean that all communication is allowed at other periods of time, since the rule is not matched at any other time.

If more than one rule matches and one or more evaluates to `allow = true`, the call continues normally (that is, all matched rule actions are set to ORed). This means that the evaluation of rules stops when the first matched rule with `allow = true` is encountered.

If more than one rule matches and all evaluate to `allow = false`, the call is barred.

If there are no matching rules, the call is not barred.





## 5 CB Service Configuration

The CB service is controlled by the **MtasCb** Managed Object (MO) and its children. An overview of the CB MO structure is shown in Figure 9.

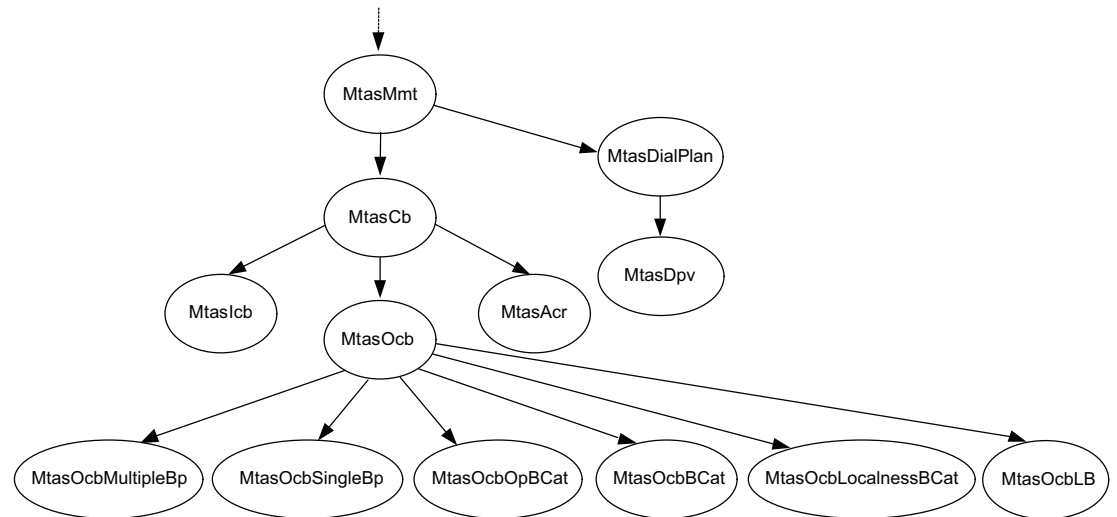


Figure 9 CB MO Structure

**Note:** All the MOs in Figure 9 can be configured on VTP level, except the **MtasDpv** MO. Meaning that this MO structure is replicable under the **VtasMmt** MO structure. The only difference is that the **VtasDialPlan** MO does not have a **VtasDpv** child.

For more information on configurable MOs and attributes related to the CB service, see [Managed Object Model \(MOM\)](#).

### 5.1 Overview Tables and Activation

This section describes the general knowledge needed for handling and activation of tables needed for configuration of the following:

- Section 5.3 Configure a User Barring Category on page 56
- Section 5.4 Configure an Operator Barring Category on page 59
- Section 5.17 Configure the Nodal Dial Plan on page 75
- Section 5.18 Configure the Per-VTP Dial Plan on page 77



### 5.1.1 View and Editing the Standby Tables Selection

In the CM browser, the MTAS offers one active and one standby view. The same MO attribute is used to display both the active and the standby table. By default the active table is presented.

The view can be changed at any time in the CM browser using the attribute `xView`, where `x` is one of `mtasDialPlan`, `mtasDpv`, `mtasOcbOpBCat`, `mtasOcbBCat`, `mtasOcb`, or `mtasIcb`, depending on the attribute. This attribute can have values `0` (Active view) or `1` (Standby view).

Editing of the tables in active view is not allowed. In standby view, however, the standby tables can be edited without affecting the traffic in any way. The changes take effect when the standby tables are activated.

Activation can be done either with immediate effect, see Section 5.1.2 Setting and Monitoring the Activation State on page 54, by setting a scheduled change time, see Section 5.1.3 Scheduling an Activation on page 55, or by using administrative operations, see Section 5.1.4 Immediate Activation of Standby Tables Using Administrative Operation on page 55. On activation the active and the standby tables are swapped. The effects of the last activation procedure can be canceled simply by repeating the activation procedure. The new entries become effective in the upcoming new sessions. Before activation, validity checks are executed on the entries.

If a configuration or activation request is rejected because of invalid data, an error with a text pointing out the reason for failure is presented to the user in the CM browser.

### 5.1.2 Setting and Monitoring the Activation State

The attribute `xActivationState` describes the activation state of the standby dial plan table and can have one of the following values:

<b>0=Idle</b>	This is the default state. There is no operation in progress.
<b>1=Activate</b>	Activation with immediate effect is requested. When the operator sets this state, the values in the standby table become active unless they are invalid. If invalid data the activation request is rejected.
<b>2=Processing</b>	A table copy operation is in progress. During this time editing the entries, changing the activation state, or entering a scheduled change time (see Section 5.1.3 Scheduling an Activation on page 55) is disabled. When the operation has finished, the state is changed automatically to <code>0=Idle</code> .



### **3=CopyToStandby**

Starts an asynchronous operation which copies the entries from the active table to the standby table. The values previously stored in the standby table are overwritten.

An activation with immediate effect can be triggered by setting `xActivationState` to `1=Activate`. While loading the data, incoming traffic requests are queued. The requests are answered when the data is fully loaded. In addition, the MTAS offers a functionality which copies the active entries to the standby table (see state `3=CopyToStandby`) as a preparation for changing the currently active entries. This operation does not affect the traffic in any way but it implicitly cancels any scheduled activation.

## **5.1.3 Scheduling an Activation**

The attribute `xChangeTime` can be used to define a time point in the future when the standby table is activated. By default `xChangeTime` is set to empty string, meaning that no change is scheduled.

The format used to specify a valid change time is: `YYYY-MM-DDThh:mm:ss` (see ISO 8601:2004(E)). For example: the value `2011-07-23T18:15:00` schedules changing the active table at 18:15:00 on 23 of July, 2011.

An activation can be scheduled only in state `0=Idle`. The execution time is limited to 2 weeks. When `xChangeTime` is set to a valid time in the future and the current standby entries are valid a change is scheduled, otherwise the configuration attempt is rejected.

The scheduled activation is handled similar to setting `xActivationState` to `1=Activate` with the only difference that the standby table does not become effective immediately but later.

A scheduled activation can be canceled by setting `xChangeTime` to the empty string at any time. Setting `xChangeTime` to a valid time in the future reschedules the activation.

While the data is loading, the activation state is changed automatically to `2=Processing`. Canceling or rescheduling activation is not possible when the loading of the data has started. When the data is fully loaded, `xChangeTime` is set back to empty string.

## **5.1.4 Immediate Activation of Standby Tables Using Administrative Operation**

The administrative operation `xActivateStandby` can be used to activate the standby tables with immediate effect. When the operator starts this operation, the values in the standby tables become active unless they are invalid. In case of invalid data, the activation request is rejected. The administrative operation cannot be started unless the value of `xActivationState` is `IDLE(0)`.



These administrative operations are supported through ECLI and NETCONF Interface.

### 5.1.5 Impact of Dropback Attribute on v<sub>tas</sub>X<sub>xx</sub> Standby and Active Tables

If the dropback attribute of a v<sub>tas</sub>X<sub>xx</sub> table changes from 1 (OTP values) to 0 (VTP values), then both the active and the standby v<sub>tp</sub>Tables is replaced with their o<sub>tp</sub>Table pair. The original values of the v<sub>tp</sub>Tables cannot be restored and the xChangeTime value is replaced. The copy is performed only if it results in a valid state otherwise the dropback attribute cannot be changed.

## 5.2 Configure ACR Display Name Evaluation

ACR evaluate reasons for lack of caller identity when determining whether a call is an anonymous call. If m<sub>tas</sub>A<sub>cr</sub>D<sub>isplay</sub>N<sub>ame</sub>E<sub>valuation</sub> in the M<sub>tas</sub>I<sub>d</sub>P<sub>res</sub> MO is set to 1 (Enabled), anonymous calls are only rejected when display-name portion of P-Asserted-Identity contains “Anonymous”. When set to 0 (Disabled) then display-name of P-Asserted-Identity is not checked by the service.

## 5.3 Configure a User Barring Category

The following sections describe how to create, modify, and delete a User Barring Category.

### 5.3.1 Create a User Barring Category

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To create a User Barring Category:

1. Navigate to the **M<sub>tas</sub>O<sub>cb</sub>** MO, see Figure 9 for its position in the MO hierarchy.
2. Right-click **M<sub>tas</sub>O<sub>cb</sub>** and click **New** in the pop-up menu.

This results in the **Set Entry Object Classes** window.

3. If there are any classes in the **Selected Classes** field, select them and click **Remove**.
4. Select **M<sub>tas</sub>O<sub>cb</sub>B<sub>C</sub>at** from the alphabetic list in the **Available Classes** field.



5. Enter the **Relative Distinguished Name (RDN)**, for example, `Mtas0cbBCat=0`, and click **Add**.

The RDN for `Mtas0cbBCat` must be an integer in the range of 0–15.

6. Click **OK**.

A new `Mtas0cbBCat` MO is presented in the CM browser.

7. In the **Table Editor** window, set the attribute `mtas0cbBCatName` to a meaningful name for this User Barring Category.

Each User Barring Category must have a unique name across all instances of User Barring Category and Operator Barring Category. No User Barring Category can have the name “Local”, “Non Local”, “L\_National”, “L\_International”, “L\_IntraLata”, “L\_IntraLataToll”, “L\_InterLata”, “L\_NanpZone1”, “L\_Nanp” and “Allow Local”.

8. Set the `mtas0cbBCatView` attribute of the MO to 1 (stand by view).

9. Click **Submit**.

10. In the **Table Editor** window, set the attribute `mtas0cbBCatBarred` to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window. To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu. This results in another row in the **Table Editor**, labeled `mtas0cbBCatBarred`.

For examples of pattern matching against the barring list, see Section 9 on page 107.

11. In the **Table Editor** window, set the attribute `mtas0cbBCatExempted` to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window. To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu. This results in another row in the **Table Editor**, labeled `mtas0cbBCatExempted`.

For examples of pattern matching against the exempted list, see Section 9 on page 107.

12. In the **Table Editor** window, do one of the following options:

- If the announcement played to calls barred by this Barring Category is to be defined by the announcement attributes of `Mtas0cb`, set the attribute `mtas0cbBCatAnnouncementControl` to 0.
- If the announcement played to calls barred by this Barring Category is to be defined by the attribute `mtas0cbBCatAnnouncementName`, set the attribute `mtas0cbBCatAnnouncementControl` to 1.



For more information, see [MTAS Announcement Management Guide](#).

13. Was `mtas0cbBCatAnnouncementControl` set to `0` in Step 12?

Yes: Continue with the next step.

No: In the **Table Editor** window, set the attribute `mtas0cbBCatAnnouncementName` to the name of a Generic Announcement, that is, an instance of `MtasGaAnn`.

For more information about Generic Announcements, see [MTAS Generic Announcement Management Guide](#).

14. In the **Table Editor** window, set the attribute `mtas0cbLocalnessBCatSSID` to the value to be reported in the Supplementary Service Identity AVP in the charging message generated when a call is barred by this Barring Category.

The meanings of the allowed values are:

<b>101</b>	Outgoing Communication Barring
<b>140</b>	National Toll Restriction
<b>141</b>	International Toll Restriction

15. Click **Submit**.

### 5.3.2 Modify a User Barring Category

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To modify a User Barring Category:

1. Navigate to the **MtasOcbBCat** MO.
2. Select the instance of `MtasOcbBCat` to be modified.
3. In the **Table Editor** window, modify the attributes as required.

To add an entry to `mtas0cbBCatBarred` or `mtas0cbBCatExempted`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtas0cbBCatBarred` or





mtasOcbBCatExempted, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an attribute, select the contents of the field to be changed and type the new value into the field.

4. Click **Submit**.

### 5.3.3 Delete a User Barring Category

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To delete a User Barring Category:

1. Navigate to the **MtasOcbBCat** MO.
2. Right-click the instance of MtasOcbBCat to be deleted, and select **Delete** in the pop-up menu.

## 5.4 Configure an Operator Barring Category

The following sections describe how to create, modify, and delete an Operator Barring Category.

### 5.4.1 Create an Operator Barring Category

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To create an Operator Barring Category:

1. Navigate to the **MtasOcb** MO, see Figure 9 for its position in the MO hierarchy.
2. Right-click **MtasOcb** and click **New** in the pop-up menu. This results in the **Set Entry Object Classes** window.
3. If there are any classes in the **Selected Classes** field, select them and click **Remove**.



4. Select **MtasOcbOpBCat** from the alphabetic list in the **Available Classes** field.
5. Enter the **Relative Distinguished Name (RDN)**, for example, **MtasOcbOpBCat=0**, and click **Add**. The RDN for **MtasOcbOpBCat** must be an integer in the range of 0–63.
6. Click **OK**.

A new **MtasOcbOpBCat** MO is presented in the CM browser.

7. Set the **mtasOcbOpBCatView** attribute of the MO to **1** (stand by view).
8. Click **Submit**.
9. In the **Table Editor** window, set the attribute **mtasOcbOpBCatName** to a meaningful name for this Operator Barring Category.

Each Operator Barring Category must have a unique name across all instances of User Barring Category and Operator Barring Category. No Operator Barring Category can have the name “Local”, “Non Local”, “L\_National”, “L\_International”, “L\_IntraLata”, “L\_IntraLataToll”, “L\_InterLata”, “L\_NanpZone1”, “L\_Nanp” and “Allow Local”.

10. In the **Table Editor** window, set the attribute **mtasOcbOpBCatNumBarred** to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window. To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu. This results in another row in the **Table Editor**, labeled **mtasOcbOpBCatNumBarred**.

For examples of pattern matching against the barring list, see Section 9 on page 107.

11. In the **Table Editor** window, set the attribute **mtasOcbOpBCatNumExempted** to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window. To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu. This results in another row in the **Table Editor**, labeled **mtasOcbOpBCatNumExempted**.

For examples of pattern matching against the exempted list, see Section 9 on page 107.

12. In the **Table Editor** window, set attribute **mtasOcbOpBCatDomain** to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window. To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu. This results in another row in the **Table Editor**, labeled **mtasOcbOpBCatDomain**.



Each string represents the host part of a non-numerical IRI to be included in this Operator Barring Category. If the first character in the string is a “\*”, this is treated as a wildcard character and a rightmost comparison of the host part of the URI is performed with the rest of the characters in the string. If the first character in the string is not a “\*”, the host part of the URI must exactly match the string.

For example \*.co.uk matches ericsson.co.uk and virgin.co.uk, virgin.co.uk only matches virgin.co.uk, and virgin\*.co.uk only matches virgin\*.co.uk.

13. In the **Table Editor** window, do one of the following options:

- If the announcement played to calls barred by this Operator Barring Category is to be defined by the announcement attributes of Mtas0cb, set the attribute mtas0cb0pBCatAnnouncementControl to 0.
- If the announcement played to calls barred by this Operator Barring Category is to be defined by the attribute mtas0cb0pBCatAnnouncementName, set the attribute mtas0cb0pBCatAnnouncementControl to 1.

For more information, see [MTAS Announcement Management Guide](#).

14. Was mtas0cb0pBCatAnnouncementControl set to 0 in Step 13?

Yes: Continue with the next step.

No: In the **Table Editor** window, set the attribute mtas0cb0pBCatAnnouncementName to the name of a Generic Announcement, that is, an instance of MtasGaAnn.

For more information about Generic Announcements, see [MTAS Generic Announcement Management Guide](#).

15. In the **Table Editor** window, set the attribute mtas0cb0pBCatSSId to the value to be reported in the Supplementary Service Identity AVP in the charging message generated when a call is barred by this Barring Category.

The meanings of the allowed values are:

<b>101</b>	Outgoing Communication Barring
<b>140</b>	National Toll Restriction
<b>141</b>	International Toll Restriction

16. Click **Submit**.



## 5.4.2 Modify an Operator Barring Category

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To modify an Operator Barring Category:

1. Navigate to the **MtasOcbOpBCat** MO.
2. Select the instance of **MtasOcbOpBCat** to be modified.
3. In the **Table Editor** window, modify the attributes as required.

To add an entry to **mtasOcbOpBCatNumBarred**, **mtasOcbOpBCatNumExempted**, or **mtasOcbOpBCatDomain** right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

Delete an entry from **mtasOcbOpBCatNumBarred**, **mtasOcbOpBCatNumExempted**, or **mtasOcbOpBCatDomain** by right-clicking the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

Modify an attribute by selecting the contents of the field to be changed and type the new value into the field.

4. Click **Submit**.

## 5.4.3 Delete an Operator Barring Category

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To delete an Operator Barring Category:

1. Navigate to the **MtasOcbOpBCat** MO.
2. Right-click the instance of **MtasOcbOpBCat** to be deleted, and select **Delete** in the pop-up menu.



## 5.5 Configure Localness Barring Categories

The nine instances of `Mtas0cbLocalnessBCat` are created by the system. None of these can be deleted, and no more can be created. The following section describes how to modify the attributes of a Localness Barring Category.

The nine instances of `Mtas0cbLocalnessBCat` have the following names: “Local”, “Non Local”, “L\_National”, “L\_International”, “L\_IntraLata”, “L\_IntraLataToll”, “L\_InterLata”, “L\_NanpZone1” and “L\_Nanp”.

The system creates each instance of `Mtas0cbLocalnessBCat` with the following attribute values:

<b>0</b>	<code>mtas0cbLocalnessBCatAnnouncementControl</code>
<b>&lt;empty&gt;</b>	<code>mtas0cbLocalnessBCatAnnouncementName</code>
<b>101</b>	<code>mtas0cbLocalnessBCatSSId</code>

### 5.5.1 Modify a Localness Barring Category

To modify a Localness Barring Category:

1. Navigate to the `Mtas0cbLocalnessBCat` MO.
2. Select the instance of `Mtas0cbLocalnessBCat` to be modified.
3. In the **Table Editor** window, modify the attributes as required.

Set the attribute `mtas0cbLocalnessBCatAnnouncementControl` to **0** if the announcement played to calls barred by this Barring Category is to be defined by the announcement attributes of `Mtas0cb`, see [MTAS Announcement Management Guide](#).

And set the attribute `mtas0cbLocalnessBCatAnnouncementControl` to **1** if the announcement played to calls barred by this Barring Category is to be defined by the attribute `mtas0cbBCatAnnouncementName`.

Set the attribute `mtas0cbLocalnessBCatAnnouncementName` to the name of a Generic Announcement, that is, an instance of `MtasGaAnn`. For more information about Generic Announcements, see [MTAS Generic Announcement Management Guide](#).

Set the attribute `mtas0cbLocalnessBCatSSId` to the value to be reported in the Supplementary Service Identity AVP in the charging message generated when a call is barred by this Barring Category. The meanings of the allowed values are:

<b>101</b>	Outgoing Communication Barring
<b>140</b>	National Toll Restriction



## 141 International Toll Restriction

4. Click **Submit**.

## 5.6 Configure Location Based Barring

The section describes how to create, modify, and delete an OCBLB.

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

### 5.6.1 Create a User Barring Category

To create a User Barring Category:

1. Navigate to the **MtasOcb** MO, see Figure 9 for its placement in the MO hierarchy.
2. Right-click **MtasOcb** and click **New** in the pop-up menu.

The **Set Entry Object Classes** window is displayed.

3. Select **MtasOcbLb** from the alphabetic list in the **Available Classes** field.

Enter the **Relative Distinguished Name (RDN)**, for example, **MtasOcbLb=0**, and click **Add**. The RDN for **MtasOcbLb** must be an integer in the range of 0–500.

4. Click **OK**.

A new **MtasOcbLb** MO is presented in the CM browser.

5. In the **Table Editor** window, set the attribute **mtasOcbLb**. This attribute is the primary key of **MtasOcbLb**. The key is in format **OCBLB<value>**, for example, **OCBLB&LOSANGELES**. Correct format is not checked by MTAS when setting attribute.
6. In the **Table Editor** window, set the attribute **mtasOcbLbNumBarred**.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window.

To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This attribute defines the leftmost parts of the global numbers and local numbers whose access is barred for the Location Based OCB. It is a list of digit



strings where each string represents one leftmost number part. For global numbers, the plus sign “+” is to be the first digit.

7. In the **Table Editor** window, set the attribute `mtasOcbLbNumExempted` to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window.

To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This attribute defines the leftmost parts of the global numbers and local numbers which are exempted from location based barring. It is a list of strings where each string represents one leftmost part of a global or local number. Each string begins with one of the strings in attribute `mtasOcbLbNumBarred`. The attribute cannot include numbers that are not already covered by attribute `mtasOcbLbNumBarred`. An exception to this advice occurs if the values in `mtasOcbLbNumBarred` and `mtasOcbLbNumExempted` must match the value of `mtasVoiceMailDepositServerAddress` or `mtasVoiceMailRetrievalServerAddress`. In this case, `mtasOcbLbNumExempted` can be set up before `mtasOcbLbNumBarred` to ensure that the checks described in dependencies for `mtasOcbLbNumBarred` and `mtasOcbLbNumExempted` are not violated.

8. In the **Table Editor** window, set the attribute `mtasOcbLbAnnouncementControl` to 0 if the announcement played to calls barred by this Barring Category needs to be defined by the announcement attributes of `MtasOcb`, see [MTAS Announcement Management Guide](#) for more information.

Set the attribute `mtasOcbLbAnnouncementControl` to 1 if the announcement played to calls barred by this Barring Category needs to be defined by the attribute `mtasOcbLbAnnouncementName`.

9. In the **Table Editor** window, set the attribute `mtasOcbLbAnnouncementName`.

In the **Table Editor** window, set the attribute `mtasOcbLbAnnouncementName` to the name of a Generic Announcement, that is, an instance of `MtasGaAnn`. For more information about Generic Announcements, see [MTAS Announcement Management Guide](#).

10. Click **Submit**.

## 5.6.2 Modify a User Barring Category

To modify a User Barring Category:

1. Navigate to the **MtasOcbLb** MO.
2. Select the instance of `MtasOcbLb` to be modified.
3. In the **Table Editor** window, modify the attributes as required.



To add an entry to `mtasOcbLbNumBarred` or `mtasOcbLbNumExempted`, right-click the attribute name and select **Add Another Value** from the pop-up menu. This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtasOcbLbNumBarred` or `mtasOcbLbNumExempted`, right-click the attribute name and select **Delete** from the pop-up menu. This results in the selected row being removed from the **Table Editor** window.

To modify an attribute, select the contents of the field to be changed and type the new value into the field.

4. Click **Submit**.

### 5.6.3 Delete a User Barring Category

To delete a User Barring Category:

1. Navigate to the **MtasOcbLb** MO.
2. Right-click the instance of **MtasOcbLb** to be deleted, and select **Delete** in the pop-up menu.

## 5.7 Configure a Barring Program in Single Scheme

This section describes how to configure a Barring Program in the Single scheme.

### 5.7.1 Create a Barring Program in Single Scheme

Before creating a Barring Program in the Single scheme, ensure that the User Barring Categories and the local Barring Categories included in the Barring Program are already defined, see Section 5.3 Configure a User Barring Category on page 56.

To create a Barring Program in the Single scheme:

1. Navigate to the **MtasOcb** MO, see Figure 9 for where it is placed in the MO hierarchy.
2. Right-click **MtasOcb** and click **New** in the pop-up menu.

This results in the **Set Entry Object Classes** window.

3. If there are any classes in the **Selected Classes** field, select them and click **Remove**.
4. Select **MtasOcbSingleBp** from the alphabetic list in the **Available Classes** field.

Enter the **Relative Distinguished Name (RDN)**, for example, `MtasOcbSingleBp=99`, and click **Add**. The RDN for `MtasOcbSingleBp` must be an integer in the range of 0–255.





5. Click **OK**.

A new `MtasOcbSingleBp` MO is presented in the CM browser.

6. If there are User Barring Categories in the Barring Program, in the **Table Editor** window, set the attribute `mtasOcbSingleBpCategories` to a list of strings. Each entry in the list of strings is shown by a separate row in the **Table Editor** window

To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu. This results in another row in the **Table Editor**, labeled `mtasOcbSingleBpCategories`. Each string is a User Barring Category number, in the range of 0–15.

7. If there are local Barring Categories in the Barring Program, in the **Table Editor** window, set the attribute `mtasOcbSingleBpLocalCats` to **1** to include the Barring Category “Local”, **2** to include the Barring Category “Non Local”, **3** to include the Barring Categories “Local” and “Non Local”, and **4** to include Barring Category “Allow Local”.
8. Click **Submit**.

## 5.7.2

### Create a Null Barring Program in Single Scheme

To support the Supplementary Service Code (SSC) for deactivating Barring Programs in the Single scheme, a Barring Program containing no User Barring Categories, and no local Barring Categories must be created, and the attribute `mtasOcbNullBarringProgram` must be set to point to that Barring Program.

To create a Barring Program in the Single scheme:

1. Navigate to the **MtasOcb** MO, see Figure 9 for where it is placed in the MO hierarchy.
2. Right-click **MtasOcb** and click **New** in the pop-up menu. This results in the **Set Entry Object Classes** window.
3. If there are any classes in the **Selected Classes** field, select them and click **Remove**.
4. Select **MtasOcbSingleBp** from the alphabetic list in the **Available Classes** field.

Enter the **Relative Distinguished Name (RDN)**, for example, `MtasOcbSingleBp=9`, and click **Add**. The RDN for `MtasOcbSingleBp` must be an integer in the range of 0–255.

5. Click **OK**.

A new `MtasOcbSingleBp` MO is presented in the CM browser.

6. Click **Submit**.



To set the Null Barring Program in the Single scheme:

1. Navigate to the **MtasOcb** MO, see Figure 9 for where it is placed in the MO hierarchy.
2. In the **Table Editor** window, set the attribute `mtasOcbNullBarringProgram` to the Relative Distinguished Name (RDN) of the `MtasOcbSingleBp` containing no Barring Categories, which were created.
3. Click **Submit**.

### 5.7.3 Modify a Barring Program in Single Scheme

To modify a Barring Program in the Single scheme:

1. Navigate to the **MtasOcbSingleBp** MO, see Figure 9 for where it is placed in the MO hierarchy.
2. Select the instance of `MtasOcbSingleBp` to be modified.
3. In the **Table Editor** window, modify the attributes as required.

To add an entry to `mtasOcbSingleBpCategories`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled `mtasOcbSingleBpCategories`.

To delete an entry from `mtasOcbSingleBpCategories`, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an attribute, select the contents of the field to be changed and type the new value into the field. In particular, set the attribute `mtasOcbSingleBpLocalCats` to define the local Barring Categories in the Barring Program; set to **1** to include the Barring Category “Local”, **2** to include the Barring Category “Non Local”, **3** to include the Barring Categories “Local” and “Non Local”, **4** to include the Barring Category “Allow Local”, and **0** to include none of the local Barring Categories.

4. Click **Submit**.

### 5.7.4 Delete a Barring Program in Single Scheme

To delete a Barring Program in the Single scheme:

1. Navigate to the **MtasOcbSingleBp** MO, see Figure 9 for where it is placed in the MO hierarchy.
2. Right-click the instance of `MtasOcbSingleBp` to be deleted, and select **Delete** in the pop-up menu.



## 5.8 Configure a Barring Program in Multiple Scheme

This section describes how to configure a Barring Program in the Multiple scheme.

### 5.8.1 Create a Barring Program in Multiple Scheme

Before creating a Barring Program in the Multiple scheme, ensure that the User Barring Categories and the local Barring Categories included in the Barring Program are already defined, see Section 5.3 Configure a User Barring Category on page 56.

To create a Barring Program in the Multiple scheme:

1. Navigate to the **MtasOcb** MO, see Figure 9 for where it is placed in the MO hierarchy.

2. Right-click **MtasOcb** and click **New** in the pop-up menu.

This results in the **Set Entry Object Classes** window.

3. If there are any classes in the **Selected Classes** field, select them and click **Remove**.
4. Select **MtasOcbMultipleBp** from the alphabetic list in the **Available Classes** field.

Enter the **Relative Distinguished Name (RDN)**, for example, **MtasOcbMultipleBp=12**, and click **Add**. The RDN for **MtasOcbMultipleBp** must be an integer in the range of 0–15.

5. Click **OK**.

A new **MtasOcbMultipleBp** MO is presented in the CM browser.

6. If there are User Barring Categories in the Barring Program, in the **Table Editor** window, set the attribute **mtasOcbMultipleBpCategories** to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window.

To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled **mtasOcbMultipleBpCategories**. Each string is a User Barring Category number, in the range of 0–15.

7. If there are local Barring Categories in the Barring Program, in the **Table Editor** window, set the attribute **mtasOcbMultipleBpLocalCats** to 1 to include the Barring Category “Local”, 2 to include the Barring Category “Non



Local”, 3 to include the Barring Categories “Local” and “Non Local”, and 4 to include the Barring Category “Allow Local”.

8. Click **Submit**.

### 5.8.2 Modify a Barring Program in Multiple Scheme

To modify a Barring Program in the Multiple scheme:

1. Navigate to the **MtasOcbMultipleBp** MO, see Figure 9 for where it is placed in the MO hierarchy.
2. Select the instance of **MtasOcbMultipleBp** to be modified.
3. In the **Table Editor** window, modify the attributes as required.

To add an entry to **mtasOcbMultipleBpCategories**, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled **mtasOcbMultipleBpCategories**.

To delete an entry from **mtasOcbMultipleBpCategories**, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an attribute, select the contents of the field to be changed and type the new value into the field. In particular, set the attribute **mtasOcbMultipleBpLocalCats** to define the local Barring Categories in the Barring Program; set to 1 to include the Barring Category “Local”, 2 to include the Barring Category “Non Local”, 3 to include the Barring Categories “Local” and “Non Local”, 4 to include the Barring Category “Allow Local”, and 0 to include none of the local Barring Categories.

4. Click **Submit**.

### 5.8.3 Delete a Barring Program in Multiple Scheme

To delete a Barring Program in the Multiple scheme:

1. Navigate to the **MtasOcbMultipleBpMO**, see Figure 9 for where it is placed in the MO hierarchy.
2. Right-click the instance of **MtasOcbMultipleBp** to be deleted, and select **Delete** in the pop-up menu.

## 5.9 Configure the Global White List

This section describes how to configure the ICB Global White List. Configuring the OCB Global White List is similar.



The ICB Global White List is defined by the following three attributes:

- `mtasIcbWhiteListNumIncl`
- `mtasIcbWhiteListNumExcl`
- `mtasIcbWhiteListDomainIncl`

The attributes `mtasIcbWhiteListNumIncl` and `mtasIcbWhiteListNumExcl` define the numerical addresses in the White List. `mtasIcbWhiteListNumIncl` lists the leftmost part of numbers to be included in the ICB Global White List. `mtasIcbWhiteListNumExcl` lists the leftmost part of numbers to be excluded from the ICB Global White List.

The attribute `mtasIcbWhiteListDomainIncl` defines the non-numerical addresses in the ICB Global White List, by listing the domains to be included in the ICB Global White List.

### 5.9.1

#### Modify List of Numbers Included in the ICB Global White List

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

For how to select and edit tables, see Section 5.1 Overview Tables and Activation on page 53.

To modify the list of numbers included in the ICB Global White List:

1. Navigate to the `MtasIcb` MO.
2. In the **Table Editor** window, modify the `mtasIcbWhiteListNumIncl` attributes as required. To add an entry to `mtasIcbWhiteListNumIncl`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtasIcbWhiteListNumIncl`, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an entry in `mtasIcbWhiteListNumIncl`, select the contents of the field to be changed and type the new value into the field.

3. Click **Submit**.



## 5.9.2 Modify List of Numbers Excluded from the ICB Global White List

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

For how to select and edit tables, see Section 5.1 Overview Tables and Activation on page 53.

To modify the list of numbers excluded from the ICB Global White List:

1. Navigate to the MtasIcb MO.
2. In the **Table Editor** window, modify the `mtasIcbWhiteListNumExcl` attributes as required. To add an entry to `mtasIcbWhiteListNumExcl`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtasIcbWhiteListNumExcl`, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an entry in `mtasIcbWhiteListNumExcl`, select the contents of the field to be changed and type the new value into the field.

3. Click **Submit**.

## 5.9.3 Modify List of Domains Included in the ICB Global White List

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

For how to select and edit tables, see Section 5.1 Overview Tables and Activation on page 53.

To modify the list of domains included in the ICB Global White List:

1. Navigate to the MtasIcb MO.
2. In the **Table Editor** window, modify the `mtasIcbWhiteListDomainIncl` attributes as required. To add an entry to `mtasIcbWhiteListDomainIncl`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.



To delete an entry from `mtasIcbWhiteListDomainIncl`, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an entry in `mtasIcbWhiteListDomainIncl`, select the contents of the field to be changed and type the new value into the field.

3. Click **Submit**.

## 5.10 Configure a Dynamic Black List

Dynamic Black List entries are inserted as ICB rules by the SSC commands for invocation of DBL and MCR, as described in *MTAS Supplementary Service Codes Management Guide*.

The identity associated with the recent call is controlled by the `mtasIcbDynamicBlackListIdSource` attribute. The maximum number of entries in the Dynamic Black List is controlled with the `mtasIcbDynamicBlackListMaxLength` attribute. The lifetime of an entry in the Dynamic Black List is controlled with the `mtasIcbDynamicBlackListEntryLife` attribute. When an entry has been in the Dynamic Black List for the duration specified by this attribute, MTAS automatically deletes the entry from the Dynamic Black List and ICB rule sets.

## 5.11 Configure an Announcement

The CB services play an audio or video announcement, or both, to indicate to the caller that barring is applied.

Announcement handling and CB announcement attributes are described in *MTAS Announcement Management Guide*.

## 5.12 Configure SIP Error Response Codes from MTAS

The `mtasMmtSendSipOrigResponse` attribute is used to set or change which SIP error response the OCB service is to send when an announcement has been played. If this attribute is set to 0, 603 (Decline), which is the normal SIP error code, is sent.

The corresponding attribute for the ICB, Anonymous Communication Rejection (ACR) and Do Not Disturb Communication Barring (DNDCB) services is `mtasMmtSendSipTermResponse`. If this attribute is set to 0, 433 (Anonymity disallowed) for ACR, 480 (Temporarily Unavailable) for DNDCB, which are the normal SIP error codes, is sent.



## 5.13 Configure a Cause Value

The MTAS can be configured to include a Q.850 cause value and corresponding cause code in a Reason header that is inserted in the SIP 183 Session Progress provisional response sent to the User Agent A, generated by the Barring services to negotiate SDP to play an announcement.

To specify the cause value for announcements defined by the attributes of `Mtas0cb`, `MtasIcb`, and `MtasAcr`, change the following attributes:

- `mtas0cbAnnCauseValue` for OCB
- `mtasIcbAnnCauseValue` for ICB, ACR, and DNDCB

If the attribute is set to `0`, no cause value is included.

To specify the cause value for announcements defined by the instances of `MtasGaAnn`, change the following attribute:

- `mtasGaAnnCauseValue`

## 5.14 Configure the Communication Barring Administrative State

The whole set of Barring services is enabled by setting the attribute `mtasCbAdministrativeState` in the `MtasCb` MO to `1` (Unlocked). If `mtasCbAdministrativeState` is set to `0` (Locked), none of the Barring services are provided.

## 5.15 Configure the Dial Plan Administrative State

The whole set of Dial Plan services is enabled by setting the attribute `mtasDialPlanAdministrativeState` in the `MtasDialPlan` MO to `1` (Unlocked). If `mtasDialPlanAdministrativeState` is set to `0` (Locked), none of the Dial Plan services are provided.

## 5.16 Configure Wholesale for Communication Barring

The CB service supports Wholesale. CB is configurable on Virtual Telephony Provider level, see Section 5.18 Configure the Per-VTP Dial Plan on page 77.

Wholesale for CB is activated when the following attributes are set to `1` (Unlocked):

- The `vtasCbAdministrativeState` attribute in the `VtasCb` MO
- The `mtasCbAdministrativeState` attribute in the `MtasCb` MO





For more information about the Wholesale service, see [MTAS Wholesale Support Management Guide](#).

## 5.17 Configure the Nodal Dial Plan

This section describes how to configure the Dial Plan that applies to all end users of the MTAS.

The Nodal Dial Plan is defined by the following three attributes:

- `mtasDialPlanAllowed`
- `mtasDialPlanExcepted`
- `mtasDialPlanDomain`

The attributes `mtasDialPlanAllowed` and `mtasDialPlanExcepted` define the numerical addresses in the Nodal Dial Plan. `mtasDialPlanAllowed` lists the leftmost part of numbers to be included in the Nodal Dial Plan. `mtasDialPlanExcepted` lists the leftmost part of numbers to be excluded from the Nodal Dial Plan.

The attribute `mtasDialPlanDomain` defines the non-numerical addresses in the Nodal Dial Plan, by listing the domains to be included in the Nodal Dial Plan.

### 5.17.1 Modify List of Numbers Included in the Nodal Dial Plan

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To modify the list of numbers included in the Nodal Dial Plan:

1. Navigate to the **MtasDialPlan** MO.
2. In the **Table Editor** window, modify the `mtasDialPlanAllowed` attributes as required.

To add an entry to `mtasDialPlanAllowed`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtasDialPlanAllowed`, right-click the attribute name and select **Delete** from the pop-up menu.



This results in the selected row being removed from the **Table Editor** window.

To modify an entry in `mtasDialPlanAllowed`, select the contents of the field to be changed and type the new value into the field.

3. Click **Submit**.

### 5.17.2 Modify List of Numbers Excluded from the Nodal Dial Plan

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

Refer to Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To modify the list of numbers excluded from the Nodal Dial Plan:

1. Navigate to the **MtasDialPlan** MO.
2. In the **Table Editor** window, modify the `mtasDialPlanExcepted` attributes as required.

To add an entry to `mtasDialPlanExcepted`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtasDialPlanExcepted`, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an entry in `mtasDialPlanExcepted`, select the contents of the field to be changed and type the new value into the field.

3. Click **Submit**.

### 5.17.3 Modify List of Domains Included in the Nodal Dial Plan

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

Refer to Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To modify the list of domains included in the Nodal Dial Plan:

1. Navigate to the **MtasDialPlan** MO.



2. In the **Table Editor** window, modify the `mtasDialPlanDomain` attributes as required.

To add an entry to `mtasDialPlanDomain`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtasDialPlanDomain`, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an entry in `mtasDialPlanDomain`, select the contents of the field to be changed and type the new value into the field.

3. Click **Submit**.

## 5.18 Configure the Per-VTP Dial Plan

This section describes how to configure the Dial Plan that applies to end users of a particular Virtual Telephony Provider.

The Per-VTP Dial Plan allows the network owner to restrict the set of addresses that end users of a particular VTP can address.

The Per-VTP Dial Plan is defined by the following three attributes:

- `mtasDpvAllowed`
- `mtasDpvExcepted`
- `mtasDpvDomain`

The attributes `mtasDpvAllowed` and `mtasDpvExcepted` define the numerical addresses in the Per-VTP Dial Plan. `mtasDpvAllowed` lists the leftmost part of numbers to be included in the Per-VTP Dial Plan. `mtasDpvExcepted` lists the leftmost part of numbers to be excluded from the Per-VTP Dial Plan.

The attribute `mtasDpvDomain` defines the non-numerical addresses in the Per-VTP Dial Plan, by listing the domains to be included in the Per-VTP Dial Plan.

### 5.18.1 Create a Per-VTP Dial Plan

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.



To create a Per-VTP Dial Plan:

1. Navigate to the **MtasDialPlan** MO, see Figure 9 for where it is placed in the MO hierarchy.
2. Right-click **MtasDialPlan** and click **New** in the pop-up menu.

This results in the **Set Entry Object Classes** window.

3. If there are any classes in the **Selected Classes** field, select them and click **Remove**.
4. Select **MtasDpv** from the alphabetic list in the **Available Classes** field.

Enter the Relative Distinguished Name (RDN), for example, **MtasDpv=12**, and click **Add**. The RDN for **MtasDpv** must be the RDN of an instance of **MtasVtp**.

5. Click **OK**.

A new **MtasDpv** MO is presented in the CM browser.

6. In the **Table Editor** window, set the attribute **MtasDpvAllowed** to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window.

To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled **MtasDpvAllowed**. Each string is the leftmost part of a set of telephone numbers that are to be allowed by this dial plan. For example, **+49** matches numbers in Germany, and **150** can match operator inquiry numbers.

7. In the **Table Editor** window, set the attribute **mtasDpvExcepted** to a list of strings.

Each entry in the list of strings is shown by a separate row in the **Table Editor** window.

To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled **mtasDpvExcepted**. Each string is the leftmost part of a set of telephone numbers that are not to be allowed by this dial plan. Each exception string begins with one of the allowed strings in this dial plan. For example, **+4912** would match premium numbers in Germany, and **150;phone-context=company.com** would match company's inquiry number.

8. In the **Table Editor** window, set the attribute **mtasDpvDomain** to a list of strings.



Each entry in the list of strings is shown by a separate row in the **Table Editor** window.

To add an entry to the list, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled `mtasDpvDomain`. Each string defines the host part of a set of non-numerical URIs that are to be allowed by this dial plan. The front part of the domain name can be wild-carded by beginning the string with `“*”`. An entry containing only a `“*”` allows access to all domains.

9. Click **Submit**.

## 5.18.2 Modify List of Numbers Included in the Per-VTP Dial Plan

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To modify the list of numbers included in the Per-VTP Dial Plan:

1. Navigate to the **MtasDpv** MO.
2. Select the instance of **MtasDpv** to be modified.
3. In the **Table Editor** window, modify the `mtasDpvAllowed` attributes as required.

To add an entry to `mtasDpvAllowed`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtasDpvAllowed`, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an entry in `mtasDpvAllowed`, select the contents of the field to be changed and type the new value into the field.

4. Click **Submit**.



### 5.18.3 Modify List of Numbers Excluded from Per-VTP Dial Plan

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To modify the list of numbers included in the Nodal Dial Plan:

1. Navigate to the **MtasDpv** MO.
2. Select the instance of **MtasDpv** to be modified.
3. In the **Table Editor** window, modify the **mtasDpvExcepted** attributes as required.

To add an entry to **mtasDpvExcepted**, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from **mtasDpvDomain**, right-click the attribute name and select **Delete** from the pop-up menu. This results in the selected row being removed from the **Table Editor** window.

To modify an entry in **mtasDpvExcepted**, select the contents of the field to be changed and type the new value into the field.

4. Click **Submit**.

### 5.18.4 Modify List of Domains Included in the Per-VTP Dial Plan

**Note:** The MTAS stores one active and one standby table for each MO. Both the active and the standby table is accessible at any time. Changing the entries is possible only in the standby table. Changes become effective for new sessions after the standby table is activated.

See Section 5.1 Overview Tables and Activation on page 53 for details on selecting and editing the tables.

To modify the list of numbers included in the Nodal Dial Plan:

1. Navigate to the **MtasDpv** MO.
2. Select the instance of **MtasDpv** to be modified.
3. In the **Table Editor** window, modify the **mtasDpvDomain** attributes as required.



4. To add an entry to `mtasDpvDomain`, right-click the attribute name and select **Add Another Value** from the pop-up menu.

This results in another row in the **Table Editor**, labeled appropriately.

To delete an entry from `mtasDpvDomain`, right-click the attribute name and select **Delete** from the pop-up menu.

This results in the selected row being removed from the **Table Editor** window.

To modify an entry in `mtasDpvDomain`, select the contents of the field to be changed and type the new value into the field.

5. Click **Submit**.

## 5.19 Configure Service Data

This section describes how to configure the service data.

### 5.19.1 Configure Operator Subscription Level Service

The operator can activate or deactivate the CB services subscription for the subscriber by setting the user data using the CAI3G protocol through the XDMS.

For more information about the CAI3G protocol, see [MTAS CAI3G Interface](#).

The following applies:

#### — Barring rules

XDMS performs the same checks on the operator rules as on the subscriber rules, as described in Section 5.19.2 Configure a Subscriber Subscription Level Service on page 82.

The check that none of the forward-to targets in the CDIV rule-set are barred by the OCB service, is supplied by CDIV and is not required to be supplied by CB.

**Note:** The check for Bar All Outgoing Calls (BAOC) by default is not performed but can be configured.

#### — Barring Programs

The operator part of the Barring Programs data in the operator part of the subscriber's XML file must comply with the operator-controlled outgoing barring programs schema, as defined in [MTAS CAI3G Interface](#).

#### — Operator Barring Programs



The Operator Controlled Barring Programs data in the operator part of the subscriber's XML file must comply with the operator-controlled outgoing barring programs schema, as defined in [MTAS CAI3G Interface](#).

— Dynamic Black List

The Dynamic Black List in the operator part of the subscriber's XML file must comply with the operator-controlled outgoing barring programs schema, as defined in [MTAS CAI3G Interface](#).

In addition to checking that a modified Dynamic Black List complies with the appropriate schema, XDMS rejects an update if any of the following checks fails:

- The content of each identity part of each identity-list element in each caller-details element is either a SIP URI or a tel URI. Each tel URI, and each SIP URI that was converted from a tel URI, contains a normalized number.

## 5.19.2 Configure a Subscriber Subscription Level Service

The subscriber data is configured through the Ut interface using the XDMS. The subscriber can configure the CB subscription rules using the conditions and action elements. The Ut interface and the XML schema for the Ut interface are described in the following documents:

— [MTAS Ut Interface](#)

— [MTAS Ut Structure](#)

The following applies:

— Barring Rules

In addition to checking that a modified rule set complies with the appropriate schema, XDMS rejects an update if any of the following checks fail:

- At most one instance of the operator-modifiable Incoming Communication Barring service data are displayed in the `simservs` document.
- At most one instance of the operator-modifiable Outgoing Communication Barring service data are displayed in the `simservs` document.
- At most one instance of the user-modifiable Incoming Communication Barring service data are displayed in the `simservs` document.
- At most one instance of the user-modifiable Outgoing Communication Barring service data are displayed in the `simservs` document.
- Only the conditions specified in Section 4.9 Subscription Rules on page 41 appear in each CB rule.





- Each CB rule contains exactly one allow action.
- At most one of the conditions `identity`, `anonymous`, and `other-identity` are displayed in any rule.
- If an identity condition contains a many `domain = host` part, and if there are except parts within that many `domain = host` part, then all those except parts are except `id = user@host`. That is, there are no except `domain` parts within a many `domain = host` part, and all the except `id = user` parts in a many `domain = host` part are users in the domain.
- The content of each one `id` part of each identity condition in an OCB rule is either a SIP URI or a tel URI.
- The content of each one `id` part of each identity condition in an ICB rule is either a SIP URI or a tel URI or a hidden: URI.
- Each hidden: URI has a counterpart entry in the `operator-dynamic-black-list` element in the operator part of the user's service data with a matching date and time.
- The content of each except `id` part of each identity condition is either a SIP URI or a tel URI.
- If an identity condition contains some one `id = <user>` parts and a many `domain = host` part, which contains some except `id = userX@host` parts, no `<user>` is displayed in both a one `id` part and an except `id` part.
- Each ICB rule contains at most one of each of the following conditions: `anonymous`, `roaming`, `international`, `international-exHC`, `identity`, `served-identity`, `other-identity`, `validity`, `invalidity`, `valid-periods`, `carrier`, and `rule-deactivated`.
- Each OCB rule contains at most one of each of the following conditions: `anonymous`, `roaming`, `international`, `international-exHC`, `identity`, `served-identity`, `other-identity`, `validity`, `invalidity`, `valid-periods`, `carrier`, `rule-deactivated`, `b-number-type`, `b-network-type`, and `localness`.
- Each tel URI in the rules, and each SIP URI in the rules that was converted from a tel URI according to Section 19.1.6 in [IETF RFC 3261](#), contains a normalized number.
- In an identity condition, each one element has a distinct `id = <user>` part.
- In an identity condition, each many element that has a `domain = <host>` part has a distinct host value.
- In an identity condition, there is at most one many element with no domain.



- In an identity condition, each except element that has an `id = <user>` part has a distinct `id` value.
- In an identity condition, each except element that has a `domain = <host>` part has a distinct `<host>` value.
- In a validity condition, each `from` element has a distinct `dateTime` value.
- In an identity condition, each except element contains either an `id = <user>` part or a `domain = <host>` part.
- In an identity condition, each `number-match` element has a distinct `normalized starts-with` part.
- If an identity condition contains some `number-match` `normalized starts-with = <partial number>` parts and a `many` part, which contains some `except id = <user>` parts, no `<user>` matches both a `number-match` `normalized starts-with` part and an `except id` part.
- Each ICB rule contains at most one of each of the following actions, `do-not-disturb`, `play-announcement` and `play-segmented-announcement`
- Each OCB rule contains at most one of each of the following actions, `play-announcement` and `play-segmented-announcement`
- If the action element contains `allow=true` and `do-not-disturb`
- If the action element contains `allow=true` and `play-announcement`
- If the action element contains `allow=true` and `play-segmentedannouncement`
- If the action element contains `play-announcement` and `play-segmented-announcement`
- The number of CB rules is less than or equal to the maximum number of rules.
- The carrier condition is activated for the user in the operator part of the OCB service, only when the user has CSRN service activated.

The check that none of the forward-to targets in the CDIV rule-set are barred by the OCB service, is supplied by CDIV and is not required to be supplied by CB.

**Note:** The check for Bar All Outgoing Calls (BAOC) by default is not performed but can be configured.

#### — Barring Programs

The subscriber part of the Barring Programs data in the subscriber's XML file must comply with the outgoing barring programs schema.



In addition to checking that a modified file complies with the appropriate schema, XDMS rejects an update if any of the following checks fail:

- If the operator part of the file contains `scheme = "single"`, then either the `subscriber` part is empty, or the `subscriber` part contains a single `program` element.
- If the operator part of the file contains `scheme = "multiple"`, then either the `subscriber` part is empty, or the `subscriber` part contains a `multiple programs` element.





## 6 Performance Management

Measurements related to the CB service are detailed in MTAS Performance Measurements.





## 7 Fault Management

The CB service has no alarms.







## 8 Barring Rule Examples

This section shows examples of rule configurations that can be applied for the CB service.

To guarantee its uniqueness, a namespace can often be a long string. XML allows a namespace to be mapped to a short string (a prefix) which makes the XML documents more readable. The mapping between each namespace and its assigned prefix as used in this section is shown in Table 3.

Table 3 Namespace Prefix Mapping

Prefix	Namespace	Purpose
cp	Urn:ietf:params:xml:ns:common-policy	Common Policies for privacy preferences as defined by the IETF
ocp	Urn:oma:xml:xm:common-policy	Common Policies for mobile as defined by OMA
ss	http://uri.etsi.org/ngn/params/xml/simservs/xcap	User Part of the MMTel document as defined by ETSI/TISPAN
mmt-op	http://schemas.ericsson.com/mmtel/operator-service-data	Operator Part of the MMTel document
mmt-serv	http://schemas.ericsson.com/mmtel/services	Ericsson defined services for inclusion in the MMTel user-data part

For more information about how the identity can be expressed, see [IETF RFC 4745](#).

### 8.1 Bar Incoming Communication from Alice

In the rule configuration shown in Example 1 the following applies for the served user:

— All incoming communication from `alice@example.com` is blocked



```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="rule1">
      <cp:conditions>
        <cp:identity>
          <cp:one id="sip:alice@example.com"/>
        </cp:identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 1 Bar Incoming Communication from Alice

## 8.2 Bar Incoming Communication from Recent Caller with Privacy

In the rule configuration shown in Example 2 the following applies for the served user:

- All incoming communication from a recent anonymous caller is blocked
- The appropriate SSC code (DBL Invocation) is submitted



```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="DBL2009-08-02T13-24-45Z">
      <cp:conditions>
        <cp:identity>
          <cp:one id="hidden:2009-08-02T13:24:45Z"/>
        </cp:identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>

<mmt-op:operator-dynamic-black-list active="true">
  <mmt-op:caller-details insertion-time="2009-08-02T13:24:45Z">
    <mmt-op:identity-list>
      <mmt-op:identity>sip:alice@example.com</mmt-op:identity>
      <mmt-op:identity>tel:+442476562000</mmt-op:identity>
      <mmt-op:identity>tel:+442476562000;ext=3024</mmt-op:identity>
    </mmt-op:identity-list>
    <mmt-op:expiry-time>2010-01-31T13:24:45Z</mmt-op:expiry-time>
    <mmt-op:reason>DBL</mmt-op:reason>
  </mmt-op:caller-details>
</mmt-op:operator-dynamic-black-list>
```

Example 2 Bar Incoming Communication from Recent Caller with Privacy

## 8.3 Bar Incoming Communication from Recent Caller without Privacy

In the rule configuration shown in Example 3, the following applies for the served user:

- All incoming communication from a recent non-anonymous caller is blocked
- The appropriate SSC code (DBL Invocation) is submitted



```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="DBL2009-08-02T13-24-45Z">
      <cp:conditions>
        <cp:identity>
          <cp:one id="sip:alice@example.com"/>
          <cp:one id="tel:+442476562000"/>
          <cp:one id="tel:+442476562000;ext=3024"/>
        </cp:identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 3 Bar Incoming Communication from Recent Caller without Privacy

## 8.4 Bar Incoming Communication from Anonymous

In the rule configuration shown in Example 4 the following applies for the served user:

- All incoming communication from an anonymous caller is blocked

The ACR function is expressed as the following rule:

- Condition: anonymous
- Action: allow=false

```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="rule2">
      <cp:conditions>
        <ss:anonymous>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 4 Bar Incoming Communication from Anonymous



## 8.5 Bar Incoming Communication from example.com except from Alice and Bob

In the rule configuration shown in Example 5 the following applies for the served user:

- All incoming communication from example.com except from Alice and Bob is blocked.

```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="rule3">
      <cp:conditions>
        <cp:identity>
          <cp:many domain="example.com">
            <cp:except id="sip:alice@example.com"/>
            <cp:except id="sip:bob@example.com"/>
          </cp:many>
        </cp:identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 5 Bar Incoming Communication from example.com except from Alice and Bob

## 8.6 Bar Incoming from Range of Numbers

The element number-match is introduced by Ericsson. In the rule configuration shown in Example 6, incoming communication from numbers starting with +4424 are barred.



```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="Coventry">
      <cp:conditions>
        <cp:identity>
          <mmt-serv:number-match starts-with="+4424"/>
        </cp:identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 6 Bar Incoming from Range of Numbers

## 8.7 Bar Incoming Communication Based on Served Identity

The call is barred when `sip:john.doe@office.com` is called but when the call arrives on some other ID of the same user, for example `sip:johnny@home.com`, the call is not barred.

```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="icb-served-id">
      <cp:conditions>
        <mmt-serv:served-identity>
          <mmt-serv:one id="sip:john.doe@office.com"/>
        </mmt-serv:served-identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 7 Bar Incoming Communication Based on Served Identity

## 8.8 Bar Outgoing Communication Based on Served Identity

The call is barred when the subscriber makes a call using the `sip:john.doe@office.com` alias PUI but when any other alias PUI of the same subscriber, for example `sip:johnny@home.com`, is used the outgoing call is allowed.



```
<ss:outgoing-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="ocb-served-id">
      <cp:conditions>
        <mmt-serv:served-identity>
          <mmt-serv:one id="sip:john.doe@office.com"/>
        </mmt-serv: served-identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:outgoing-communication-barring>
```

Example 8 Bar Outgoing Communication Based on Served Identity

## 8.9 Bar Outgoing Communication Based on Subscriber-state Except 1800 Pre-Paid Calls

When the subscriber is making a call, the MTAS checks the current subscriber-state value and the operator-configured rule for the subscriber-state.

In the rule configuration shown in Example 9, outgoing communication is barred, except for 1800 pre-paid calls, after playing an announcement (if configured) to the served user. This applies when the subscriber-state value HalfSuspension in the common data and the operator-configured outgoing barring rule subscriber-state=HalfSuspension matches.



```

<mmt-data:operator-configuration>
  <mmt-data:operator-service-data>
    <mmt-op:operator-outgoing-communication-barring activated="true">
      <cp:ruleset>
        <cp:rule id="Halfsuspension">
          <!-- This takes care of barring all OG
                call in half Suspension state -->
          <cp:conditions>
            <mmt-serv:subscriber-state>HalfSuspension
          </mmt-serv:subscriber-state>
          </cp:conditions>
          <cp:actions>
            <ss:allow>false</ss:allow>
            <mmt-serv:play-announcement>
              HalfSuspensionAnn
            </mmt-serv:play-announcement>
          </cp:actions>
        </cp:rule>
        <cp:rule id="halfsuspension_allow_prepaid">
          <!-- This takes care of allowing prepaid
                OG call in half Suspension state -->
          <cp:conditions>
            <mmt-serv:subscriber-state>HalfSuspension
          </mmt-serv:subscriber-state>
            <cp:identity>
              <mmt-serv:number-match starts-with="+1800"/>
            </cp:identity>
          </cp:conditions>
          <cp:actions>
            <ss:allow>true</ss:allow>
          </cp:actions>
        </cp:rule>
      </cp:ruleset>
    </mmt-op:operator-outgoing-communication-barring>
  </mmt-data:operator-service-data>
  <mmt-op:operator-common-data>
    <mmt-op:subscriber-state>HalfSuspension
  </mmt-op:subscriber-state>
  </mmt-op:operator-common-data>
</mmt-data:operator-configuration>

```

Example 9 Bar Outgoing Communication Based on Subscriber-state Except 1800 Pre-paid Calls

## 8.10 Bar Incoming Communication Based on Subscriber-state

When the subscriber has an incoming call, the MTAS checks the current subscriber-state value and operator configured rule for the subscriber-state.

In the rule configuration shown in Example 10, the incoming call is barred after playing an announcement (if configured) to the served user. This applies





when the subscriber-state value TotalSuspension in the common data and the operator-configured incoming barring rule subscriber-state=TotalSuspension matches.

```
<mmt-data:operator-configuration>
  <mmt-data:operator-service-data>
    <mmt-op:operator-incoming-communication-barring activated="true">
      <cp:ruleset>
        <cp:rule id="totalsuspension">
          <cp:conditions>
            <mmt-serv:subscriber-state>
              TotalSuspension
            </mmt-serv:subscriber-state>
          </cp:conditions>
          <cp:actions>
            <ss:allow>false</ss:allow>
            <mmt-serv:play-announcement>
              TotalSuspensionAnn
            </mmt-serv:play-announcement>
          </cp:actions>
        </cp:rule>
      </mmt-op:operator-incoming-communication-barring>
    </mmt-data:operator-service-data>

    <mmt-op:operator-common-data>
      <mmt-op:subscriber-state>
        TotalSuspension
      </mmt-op:subscriber-state>
    </mmt-op:operator-common-data>
  </mmt-data:operator-configuration>
```

Example 10 Bar Incoming Communication Based on Subscriber-state

## 8.11 White List

In the rule configurations shown in this section, the same white list is expressed in two variants. In Example 11, the rule is expressed with the element `many` together with `except`. Meanwhile, in Example 12, the rule is expressed with the element `ocp:other-identity`.

The following applies for the served user:

- Only bob@good.example.net and +12125551234 are allowed to establish an incoming communication



```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="rule4">
      <cp:conditions>
        <cp:identity>
          <cp:many>
            <cp:except id="sip:bob@good.example.net"/>
            <cp:except id="tel:+12125551234"/>
          </cp:many>
        </cp:identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 11 White List Variant One

```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="rule6">
      <cp:conditions>
        <cp:identity>
          <cp:one id="sip:bob@good.example.net"/>
          <cp:one id="tel:+12125551234"/>
        </cp:identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>true</ss:allow>
      </cp:actions>
    </cp:rule>
    <cp:rule id="rule7">
      <cp:conditions>
        <ocp:other-identity/>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 12 White List Variant Two

## 8.12 Conditions Combined in One Rule and Different Rules

This section describes how different conditions can be combined and use precedence and the action for incoming communication to achieve different semantics.



The ACR, when the anonymous condition is included in one rule, and validity are combined as follows:

- Bar all communication between 10.00–11.00 2005-12-14 and always bar anonymous.
- Bar anonymous communication between 10.00–11.00 2005-12-14.
- Allow all communication between 10.00–11.00 2005-12-14 and bar anonymous the other period time.

In the rule configuration shown in Example 13, validity and anonymous are expressed in separate barring rules. If an anonymous communication is received during the time specified in the validity condition, the communication is barred since both rule 1 and rule 2 matches, and the aggregated action is as follows:

```

validity=true
AND
allow=false
OR
anonymous=true
AND
allow=false

```

```

<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="rule1">
      <cp:conditions>
        <cp:validity>
          <cp:from>2005-12-14T10:00:00.000+01:00</cp:from>
          <cp:until>2005-12-14T11:00:00.000+01:00</cp:until>
        </cp:validity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
    <cp:rule id="rule2">
      <cp:conditions>
        <ss:anonymous/>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>

```

**Example 13** Validity and Anonymous in Separate Barring Rules



In the rule configuration shown in Example 14, validity and anonymous are expressed in one barring rule. If a non-anonymous communication is received during the time specified in the validity condition, rule 1 evaluates to the following:

```
validity=true  
AND  
anonymous=false
```

Since there are no more rules, no rule matches and the communication is allowed.

```
<ss:incoming-communication-barring active="true">  
  <cp:ruleset>  
    <cp:rule id="rule1">  
      <cp:conditions>  
        <cp:validity>  
          <cp:from>2005-12-14T10:00:00.000+01:00</cp:from>  
          <cp:until>2005-12-14T11:00:00.000+01:00</cp:until>  
        </cp:validity>  
        <ss:anonymous/>  
      </cp:conditions>  
      <cp:actions>  
        <ss:allow>false</ss:allow>  
      </cp:actions>  
    </cp:rule>  
  </cp:ruleset>  
</ss:incoming-communication-barring>
```

#### Example 14 Validity and Anonymous in One Barring Rule

In the rule configuration shown in Example 15, validity is given in an allowing rule and anonymous in a barring rule. If an anonymous communication is received during the time specified in the validity condition, the communication is allowed since both rule 1 and rule 2 matches, and the aggregated action is as follows:

```
validity=true  
AND  
allow=true  
OR  
anonymous=true  
AND  
allow=false
```



```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="rule1">
      <cp:conditions>
        <cp:validity>
          <cp:from>2005-12-14T10:00:00.000+01:00</cp:from>
          <cp:until>2005-12-14T11:00:00.000+01:00</cp:until>
        </cp:validity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>true</ss:allow>
      </cp:actions>
    </cp:rule>
    <cp:rule id="rule2">
      <cp:conditions>
        <ss:anonymous/>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 15 Validity in Allowing Rule and Anonymous in Barring Rule

## 8.13 Playing Generic Announcement – play-announcement

Barring incoming communication from `alice@example.com` and playing operator-named announcement “Call Me Later” to the calling party.

```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="generic-announcement">
      <cp:conditions>
        <cp:identity>
          <cp:one id="sip:alice@example.com"/>
        </cp:identity>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
        <mmt-serv:play-announcement>Call Me Later</mmt-serv:play-announcement>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 16 Bar Incoming Communication from Alice and Play Generic Announcement



## 8.14 Playing Generic Announcement – play-segmented-announcement

Barring incoming communication unconditionally and playing operator-named segmented announcement including several announcement voice variables to the calling party.

```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="relocation">
      <cp:conditions/>
      <cp:actions>
        <ss:allow>false</ss:allow>
        <mmt-serv:play-segmented-announcement announcement-name="reloc_ann">
          <mmt-serv:announcement-variable variable-name="NewNumberAreaCode">
            <mmt-serv:variable-value>0211</mmt-serv:variable-value>
          </mmt-serv:announcement-variable>
          <mmt-serv:announcement-variable variable-name="NewLocalNumber">
            <mmt-serv:variable-value>3811973 </mmt-serv:variable-value>
          </mmt-serv:announcement-variable>
        </mmt-serv:play-segmented-announcement>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 17 Bar Incoming Communication Unconditionally and Play Generic Segmented Announcement

## 8.15 Do Not Disturb Communication Barring

These are examples of Do Not Disturb Communication Barring rules including the rule example of the hybrid service with ACR.

```
<ss:incoming-communication-barring active="true">
  <cp:ruleset>
    <cp:rule id="hybrid-dndcb-acr">
      <cp:conditions>
        <ss:anonymous/>
      </cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
        <mmt-serv:do-not-disturb/>
      </cp:actions>
    </cp:rule>
    <cp:rule id="DND CB">
      <cp:conditions></cp:conditions>
      <cp:actions>
        <ss:allow>false</ss:allow>
        <mmt-serv:do-not-disturb/>
      </cp:actions>
    </cp:rule>
  </cp:ruleset>
</ss:incoming-communication-barring>
```

Example 18 Bar Incoming Communication with Do Not Disturb



## 8.16 Bar Outgoing Communication Based on Call Analysis

When the subscriber is making a call, MTAS will match B-Number classification returned by Number Translation Service, B-Number network-type returned by Number Portability Service and localness value returned by Number Analysis Component with the operator-configured conditions b-number-type, b-network-type, and localness.

In the rule configuration shown in below example, outgoing communication is barred when the rule is matched and an announcement is played. The call is rejected with error status code = 487 and Reason Header with reason-protocol = Q.850, reason-cause=127, and reason-text="Interworking".

```
<mmt-data:operator-configuration>
  <mmt-data:operator-service-data>
    <mmt-op:operator-outgoing-communication-barring activated="true">
      <cp:ruleset>
        <cp:rule id="rule1">
          <cp:conditions>
            <mmt-serv:b-number-type>GP4</mmt-serv:b-number-type>
            <mmt-serv:b-network-type>Fixed</mmt-serv:b-network-type>
            <mmt-serv:localness>Non Local</mmt-serv:localness>
          </cp:conditions>
          <cp:actions>
            <ss:allow>false</ss:allow>
            <mmt-serv:play-announcement>Dial01AreaCodeTelephoneNumber
            </mmt-serv:play-announcement>
            <mmt-serv:status-code>487</mmt-serv:status-code>
            <mmt-serv:reason-header>
              <mmt-serv:reason-protocol>Q.850</mmt-serv:reason-protocol>
              <mmt-serv:reason-cause>127</mmt-serv:reason-cause>
              <mmt-serv:reason-text>Interworking</mmt-serv:reason-text>
            </mmt-serv:reason-header>
          </cp:actions>
        </cp:rule>
      </cp:ruleset>
    </mmt-op:operator-outgoing-communication-barring>
  </mmt-data:operator-service-data>
</mmt-data:operator-configuration>
```

**Example 19** Bar Outgoing Communication Based on Call Analysis







## 9 OCB Barred and Exempted List Configuration Examples

This section gives examples where the barring and exempted lists of the Operator Barring Category or Barring Category are configured to match against wildcard and fixed length patterns in list entries.

Each list entry is the part furthest to the left of a telephone number that is to either be included or exempted from inclusion in the specified Barring Category. Entries are prefix matched against the entries in the list of numbers to be barred, and prefix or fixed length matched against entries in the list of numbers to be exempted. For example, +447 can be matched against mobile numbers in the UK, and 150 can be matched against operator inquiry numbers. Each exemption entry begins with one of the barred entries in the Barring Category.

An entry can include one or more wildcard characters (^) at any position in the number interpreted as character values. For example, the entry 30^^;phone-context=+33 in the barred list matches tel:3044;phone-context=+33.

An entry can include the \$ character at the last position for fixed length number matching. For example, the entry 30^^;phone-context=+33\$ in the exempted list matches tel:3044;phone-context=+33 and 1234^\$ matches the URI 12345@example.com.

### 9.1 Wildcard Pattern in List Entry

#### 9.1.1 Local Number

In Table 4, the results of matching a local number with a wildcard pattern in a list entry against different numbers is explained. The pattern matching is done with the following conditions:

- Barred number: 3
- Exempted number: 30^^;phone-context=+33

The matching pattern 30^^ is allowed as an entry in both the exempted and barred list. For a string to prefix match with 30^^, it must have a minimum of 4 characters and start with 30 followed by two digits, which means that prefixes that match with 30^^ start with numbers in the range of 3000–3099.



Table 4 Wildcard Pattern in List Entry with Local Number

Number Type	Normalized R-URI, User Part	Result
Local number	30445;phone-context=+33	<ul style="list-style-type: none"><li>• The number is barred.</li><li>• The left string matches in the barred list with 3.</li><li>• There is no match in the exempted list as the number length is 5 digits.</li></ul>
Local number	3044;phone-context=+3	<ul style="list-style-type: none"><li>• The number is not barred.</li><li>• The left string matches in the barred list with 3.</li><li>• There is a match in the exempted list as the number length is 4 digits, 44 is matched as the wildcard, and the phone-context is identical.</li></ul>
Local number	3055;phone-context=+33	<ul style="list-style-type: none"><li>• The number is not barred.</li><li>• The left string matches in the barred list with 3.</li><li>• There is a match in the exempted list as the number length is 4 digits, 55 is matched as the wildcard, and the phone-context is identical.</li></ul>
Local number	3044;phone-context=+36	<ul style="list-style-type: none"><li>• The number is barred.</li><li>• The left string matches in the barred list with 3.</li><li>• There is no match in the exempted list as the phone-context, +36, is different.</li></ul>



Number Type	Normalized R-URI, User Part	Result
Local number	4044;phone-context=+33	<ul style="list-style-type: none"> <li>• The number is not barred.</li> <li>• There is no match in the barred list with 4044.</li> <li>• There is no match in the exempted list.</li> </ul>
Global number	+36302119345	<ul style="list-style-type: none"> <li>• The number is not barred.</li> <li>• There is no match in the barred list with +3630.</li> <li>• There is no match in the exempted list.</li> </ul>

### 9.1.2

#### Global Number

In Table 5, the results of matching a global number with a wildcard pattern in a list entry against different numbers is explained. The pattern matching is done with the following conditions:

- Barred number: +33^^1
- Exempted number: +3360123

Table 5 Wildcard Pattern in List Entry with Global Number

Number Type	Normalized R-URI, User part	Result
Local number	3344	<ul style="list-style-type: none"> <li>• The number is not barred.</li> <li>• There is no match in the barred list with 3344.</li> <li>• There is no match in the exempted list.</li> </ul>
Global number	+337012345678	<ul style="list-style-type: none"> <li>• The number is barred.</li> <li>• The left string matches in the barred list with +33701 and 70 is matched as the wildcard.</li> <li>• There is no match in the exempted list.</li> </ul>



Number Type	Normalized R-URI, User part	Result
Global number	+336011345678	<ul style="list-style-type: none"><li>• The number is barred.</li><li>• The left string matches in the barred list with +33601 and 60 is matched as the wildcard.</li><li>• There is no match in the exempted list.</li></ul>
Global number	+336012345678	<ul style="list-style-type: none"><li>• The number is not barred.</li><li>• The left string matches in the barred list with +33601 and 60 is matched as the wildcard.</li><li>• The left string matches in the exempted list with +3360123.</li></ul>
Global number	+336031193456	<ul style="list-style-type: none"><li>• The number is not barred.</li><li>• There is no match in the barred list with +33603.</li><li>• There is no match in the exempted list.</li></ul>

## 9.2 Fixed Length Pattern in List Entry

### 9.2.1 Local Number

In Table 6, the results of matching a local number with a fixed length pattern in a list entry against different numbers is explained. The pattern matching is done with the following conditions:

- Barred number: 3
- Exempted number: 123;phone-context=+33\$



Table 6 Fixed Length Pattern in List Entry with Local Number

Number Type	Normalized R-URI, User Part	Result
Local number	123;phone-context=+334	<ul style="list-style-type: none"> <li>• The number is barred.</li> <li>• The left string matches in the barred list with 3.</li> <li>• There is no match in the exempted list as +334 is longer than the fixed length limit +33\$.</li> </ul>
Local number	123;phone-context=+33	<ul style="list-style-type: none"> <li>• The number is not barred.</li> <li>• The left string matches in the barred list with 3.</li> <li>• There is a match in the exempted list and the phone-context is identical.</li> </ul>
Local number	123;phone-context=+3	<ul style="list-style-type: none"> <li>• The number is barred.</li> <li>• The left string matches in the barred list with 3.</li> <li>• The phone-context does not match in the exempted list.</li> </ul>

## 9.2.2

### Short Number

In Table 7, the results of matching a short number with a fixed length pattern in a list entry against different numbers is explained. The pattern matching is done with the following conditions:

- Barred number: 3
- Exempted number: 30^^\$



Table 7 Fixed Length Pattern in List Entry with Short Number

Number Type	Normalized R-URI, User Part	Result
Local number	30445	<ul style="list-style-type: none"><li>• The number is barred.</li><li>• The left string matches in the barred list with 3.</li><li>• There is no match in the exempted list as 30445 is longer than the fixed length limit 30^^\$.</li></ul>
Local number	3044	<ul style="list-style-type: none"><li>• The number is not barred.</li><li>• The left string matches in the barred list with 3.</li><li>• There is a match in the exempted list as 3044 matches the fixed length limit and 44 is matched as the wildcard.</li></ul>
Local number	3055	<ul style="list-style-type: none"><li>• The number is not barred.</li><li>• The left string matches in the barred list with 3.</li><li>• There is a match in the exempted list as 3055 matches the fixed length limit and 55 is matched as the wildcard.</li></ul>
Local number	3144	<ul style="list-style-type: none"><li>• The number is barred.</li><li>• The left string matches in the barred list with 3.</li><li>• There is no match in the exempted list with 3144 .</li></ul>



Number Type	Normalized R-URI, User Part	Result
Local number	4044	<ul style="list-style-type: none"><li>• The number is not barred.</li><li>• There is no match in the barred list with 4044.</li><li>• There is no match in the exempted list.</li></ul>
Global number	+33602119345	<ul style="list-style-type: none"><li>• The number is not barred.</li><li>• There is no match in the barred list with +3360.</li><li>• There is no match in the exempted list.</li></ul>