

MTAS Session Transfer to Own Device Management Guide

MTAS

USER GUIDE

Copyright

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

Disclaimer

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

Trademark List

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



Contents

1	Introduction	1
1.1	Prerequisites	1
2	Overview	3
2.1	STOD Call Push	3
2.2	STOD Call Pull	5
2.3	Interaction with Other Services	6
3	Subscription Rules	15
3.1	Conditions	16
3.2	Actions	20
4	STOD Configuration	23
4.1	STOD Administrative State Configuration	23
4.2	Call Pull Configuration	24
4.3	Wholesale for STOD Configuration	25
4.4	Session Timer Configuration	25
4.5	Service Data Configuration	25
4.6	Reason-text Configuration in BYE Message	27
5	Prerequisites for Full Operation of Call Pull Replaces	29
6	Performance Management	31
7	Fault Management	33
8	STOD Rule Examples	35
8.1	Unconditional Parallel Ringing	35
8.2	Time-based Parallel Ringing	35
8.3	Serial Ringing Consecutive Days, Valid Periods	36





1 Introduction

This document describes how to configure the Session Transfer to Own Device (STOD) service in the MTAS.

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 the STOD service, the STOD license must be installed. The STOD service is dependent of the Application Server (AS) Controlled Forking which must have a valid license for the STOD service to be enabled.

For more information about the licenses, refer to *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 STOD service is an IMS service which allows an originating or terminating served user to transfer an ongoing communication session from one device to another device belonging to the same user (terminals defined in the subscription data) or to a related user.

The STOD service consists of two subservices: STOD Call Push and STOD Call Pull.

2.1 STOD Call Push

When the Call Push service is triggered by the served user, the communication leg to the served user is terminated, and the serving MTAS initiates serial, parallel, or flexible session establishment to the target devices or related users, or both. The communication session is transferred to the target first answering the call.

The list of devices belonging to the served user and related users contains up to 10 terminal identities or related users identities, or both. It is configured by the network operator or user.

The served user can set rules through the Ut interface. These rules can be defined for parallel, serial, and flexible modes of ringing. The rules can have day and time defined, controlling the day and time when the rules are applied. The served user can also define the duration allowed before answer for targets in parallel and serial modes.

While the parallel, serial, or flexible distribution is being attempted the other party can, subject to configuration, be played a Communication Hold announcement.

Communication setup attempts to the IMS Primary User identity are handled as IMS originating or terminating calls.

When more than one device of the IMS Primary User is included in the rules, each INVITE is routed as an originating or terminating call, with the corresponding provisioned terminal selector, but with different Call-IDs in the incoming leg.

Communication setup attempts to Related User identities are treated as new outgoing call requests. No prefixes are used in the outgoing INVITE Requests to these identities. Related User identities can be in the IMS network or any other network. Attempts by the served user to define rules based on conditions not supported by the Call Push service are rejected by XML Document Management Server (XDMS).



2.1.1 Serial, Parallel, and Flexible Distribution

Serial distribution calls each of the targets identified in the active rule consecutively until one target answers or all targets have been unsuccessfully attempted. Each communication setup attempt is limited in duration by an answer timer. Following expiry of the timer the communication setup attempt is canceled and the next target identity in the active rule, if any, is attempted.

Parallel distribution calls all targets in the active rule at the same time until one target answers or the time allowed for an answer expires. Following answer or timer expiry communication setup attempts to unanswered targets from which a final response has not been received are canceled.

Flexible distribution combines serial and parallel distribution, that is some of the targets are called serially and some in parallel. The target list is processed from the top to the bottom. If a target is marked with serial, it is rung simultaneously with the next parallel marked targets below this target until a target marked with serial is found, or the end of the target list is reached. If a target is marked with parallel, it is rung simultaneously with the previous serial target and with the similarly marked targets below this target until a target marked with serial is found, or the end of the target list is reached.

2.1.2 Controlled Forking

When the target is a device from the User Common Data, then a new `Accept-Contact` header is appended in the `INVITE` sent to the IMS Primary User. The feature preference in the `Accept-Contact` header is constructed from the provisioned terminal selector, and from the terminal selector prefix specified by the CM attribute `mtasMmtTerminalSelectorPrefix`. The “require” and “explicit” parameters are included in the `Accept-Contact` header as follows:

```
Accept-Contact: *;<prefix><selector>;require;explicit
```

As the `Accept-Contact` header can already contain feature preferences in incoming initial `INVITE`, adding new feature preferences in `Accept-Contact` header by the STOD service leads to collisions. The `mtasFcdCallerPrefReqFilter` attribute defines which feature preferences of the `Accept-Contact` and `Reject-Contact` headers of an incoming request are not to be copied by the STOD service to outgoing requests towards distribution targets to avoid collisions with feature preferences added by the STOD service itself. It contains a list of feature tags, for example, “mobility”. An empty string, being a default value of the `mtasFcdCallerPrefReqFilter` attribute, means that all caller preferences, besides `sip.instance` is copied.

The `sip.instance` feature preference is always filtered out independently of this setting. Parameter configured as terminal selector prefix (`mtasMmtTerminalSelectorPrefix`) is to be typically added to the list of filtered out feature preferences.



A valid AS Controlled Forking License must be present for using the common device list.

For more information about controlled forking, refer to *MTAS Target Handling Management Guide*.

2.1.3 Subfunctions

The subfunctions included in the STOD service are described in this section.

2.1.3.1 Ring Targets

This subfunction is responsible for the serial or parallel ringing of the users, specified in the served XML data of the user.

2.1.3.2 Auto-Answer Avoidance

Auto-Answer Avoidance is responsible for requesting a confirmation from the answering user before FCD classifies the call to have been answered. This is to avoid an automate device, like voicemail, from answering the call and thus stopping ringing on other devices. This function is only applicable to related user targets.

If there is no DTMF confirmation, the call leg is treated as if it was rejected by the called party:

- In case of the serial distribution the next target is attempted (if the called user was the last one in the target list, the communication is rejected).
- In case of the parallel distribution other targets are not affected and they continue ringing.

The use of the Auto-Answer Avoidance function is controlled by `mtasAutoAnswerAvoidanceAnnouncement` configuration parameter and by `auto-answer-avoidance` and `auto-answer-avoidance-condition` elements.

2.2 STOD Call Pull

Call Pull is a feature which allows transfer of established communication session between devices belonging to the same subscriber on subscriber request. The subscriber is able to transfer established call to another device by sending a SIP request with dialog replacement information or a special service code from the “Pull” initiating device. If all preconditions for Call Pull are fulfilled, MMTel AS seamlessly transfers the communication session.



2.3 Interaction with Other Services

This section describes how the Call Push and Call Pull services interacts with other services.

2.3.1 Advice of Charge

Credit warning and credit limit reached announcements during a transferred session is played to the credit owner only if the user is involved in the transferred call. Those cases are handled as normal mid-session and session termination credit announcements.

When a Call Push user that subscribes to Advice of Charge (AoC) transfers a session to a Related User, no AoC information is sent to the Related User. The MTAS does not initiate an AoC Session with the OCS for the originating charging session to Related User.

When a Call Push user that subscribes to AoC transfers a session to a Primary user, AoC information is continued to be sent to the served user, if the terminal supports the AoC MIME Type and if the SIP INFO method is allowed.

2.3.2 Ad-Hoc Conference

Originating Call Push

An IMS user can use the Call Push service, as an originating Conference Creator and transfer an Ad-Hoc Conference to a Primary or Related user. The “isfocus” feature parameter is forwarded to the Primary or Related user.

An IMS user cannot use the Call Push service, as an originating Conference Creator and transfer a session in a Co-located Ad-Hoc Conference. If Call Push is started, the Conference is terminated.

Terminating Call Push

An IMS user can use the Call Push service as a terminating Conference Participant and transfer a session in a Standalone or Co-located Ad-Hoc Conference to a Primary or Related user. The “isfocus” feature parameter is forwarded to the Primary or Related user.

Call Pull

An IMS user cannot pull a conference session when being the Conference Creator in a Co-located Ad-Hoc Conference. The Call Pull attempt is in this case rejected with error code 403 and informative warning text.



Call Pull of conference sessions where the served user is an ordinary conference participant is supported. Any active conference event subscription is terminated when Call Pull takes place.

2.3.3 Call Admission Control

The User and Group Call Admission Control counters are not affected when Call Push is started.

Originating Call Push

A Call Push call counts as a single originating call for both User and Group Call Admission Control.

Terminating Call Push

A Call Push call counts as a single terminating call for both User and Group Call Admission Control.

Call Pull

The User Call Admission Control employs predictive algorithm to determine if a successful Call Pull started through the Replaces header would reach the UCAC limits and can reject the pull request.

Call Pull requests started through the Supplementary Service Code (SSC) are counted as a new session by the Call Admission Control service.

2.3.4 Carrier Pre-Select and Carrier Pre-Select Rn

Initial `INVITE` requests sent by the Call Push service to Related Users are subject to the Carrier Pre-Select or Carrier Pre-Select Rn service.

For more information about the CPS services, refer to *MTAS Carrier Select and Carrier Pre-Select Management Guide*.

2.3.5 Carrier Select and Carrier Select Rn

The Call Push service allows the Related User identity to include a Carrier Select (CS) code.

The Related Users are subject to Congestion Control in Carrier Select Rn, which means that the CrankBack service can intercept an error response from the user and if it has a specific Q.850 code, CrankBack repeats the call.

For more information about the CS services, refer to *MTAS Carrier Select and Carrier Pre-Select Management Guide*.



2.3.6 Charging

For information on the interaction between Call Push and offline charging, refer to *Diameter Offline Charging in MTAS*.

For information on the interaction between Call Push and online charging, refer to *Diameter Online Charging in MTAS*.

2.3.7 Communication Barring

Originating/Terminating Call Push

The Outgoing Communication Barring (OCB) service is performed on the outgoing `INVITE` requests to the Related Users.

Related User identities cannot be an identity which would be barred by OCB or included in the Communication Diversion (CDIV) blacklist.

Call Pull

Call Pull requests are monitored for a tentative roaming restriction of the served user.

2.3.8 Communication Completion

If Communication Hold and SIP BYE is started for an originating session after the Communication Completion (CC) service has been completed, the Originating Call Push service is not executed. Instead, the session is terminated.

2.3.9 Communication Diversion

Originating/Terminating Call Push

The CDIV service is not started when the Call Push service transfers the session to a Primary User.

Call Pull

Call Pull request for diverted sessions by the served user is rejected by the Call Pull service.

Call Pull request for diverted sessions by other party can be allowed by setting the appropriate policy, CM parameter `mtasStodCallPullPolicyDiversion`.



2.3.10 Dialed Number Mapping

The Dialed Number Mapping (DNM) service is not started after the Call Push is started and distributes the call. Therefore, the seven-digit and short-code number must never be used as the target of Call Push service.

2.3.11 ECT

Call Pull is not available for transferred sessions initiated by the served user.

2.3.12 Emergency State

Call Pull requests from users in Emergency Callback Window state can be allowed by setting the appropriate policy, CM parameter `mtasStodCallPullPolicyEmCbw`.

2.3.13 Flexible Communication Distribution

The Call Push service uses the Flexible Communication Distribution (FCD) service to perform distribution in accordance with the defined rule set; this independently of the administrative state of the FCD service.

The FCD to Primary User's Devices function is also applicable to Call Push initiated distributions, if this FCD option is enabled.

The FCD - Divert Primary (FCDDP) option is not started at Call Push initiated distribution.

When the Call Push service is active but no Call Push-specific ruleset is provisioned to the served user, the Call Push service uses the FCD-specific ruleset, if provisioned.

2.3.14 Flexible Service Format Selection

The Call Push service can be suppressed using the FSFS service. The MTAS starts the FSFS service before the Call Push service when receiving the SIP INVITE.

The FSFS service provides an indicator for the MTAS to suppress the Call Push service. When the FSFS service suppresses the Call Push service, the SIP BYE that can trigger the Call Push service to distribute the communication, is processed as if the service was not active. The MTAS simply forwards the SIP BYE to the caller and so the communication session is released.

For more information about the FSFS service, refer to *MTAS Flexible Service Format Selection Management Guide*.



2.3.15 Handling of OPTIONS

The Call Push service handles the in-dialog SIP OPTIONS request.

When in-dialog SIP OPTIONS requests are received on the leg of the originating side (Terminating Call Push) or the terminating side (Originating Call Push) during Call Push invocation, MTAS sends a “200 OK” response.

No SDP is included in the response.

2.3.16 Handling of Session Refresh Messages when Session Timer Procedures are Suppressed

It is possible to configure MTAS to suppress Session Timer procedures for UE sessions, by setting the CM attribute `mtasSipUeSessionTimerSupport` to disabled.

When session refresh requests are received on the leg of the originating side, (Terminating Call Push) or the terminating side (Originating Call Push) during Call Push invocation, MTAS sends a “200 OK” response. The 200 OK response does not contain any session refresh related headers or parameters. In case of session refreshing re-INVITE, an SDP body is included in the 200 OK response. The body contains the last negotiated local SDP of the given leg, with the same o-line.

2.3.17 Identity Presentation

This section describes the Call Push interactions with Identity Presentation services.

For more information about the Identity Presentation services, refer to *MTAS Identity Presentation Management Guide*.

Dynamic Ad-Hoc Identity Presentation

The Call Push service allows the Related User identity to include an SSC code indicating Ad-hoc Identity Restriction or Presentation in the identities in the target list.

Originating Identity Presentation

Terminating Call Push

When the target is a Primary User, the Originating Identity Presentation (OIP) service operates on all messages towards the terminating entity.



Originating Identity Restriction

Terminating and Originating Call Push

If the served user has OIR settings which provide privacy for the user identity, then in Call Push outgoing INVITE requests to a Related User, the OIR service performs the following actions:

- Escape the Privacy header “history” within the hi-targeted-to-uri field (in the History Info header) containing the served user’s identity in the INVITE message
- Change the To-header to the target’s URI

The same restrictions are applied to all other messages containing these headers in the same direction as the initial INVITE request.

Originating Call Push

A `P-Asserted-Identity` header field received by the Call Push AS in a final response from a Primary or Related user is passed unmodified in the `UPDATE` message to the terminating entity.

A `Privacy` header field received by the Call Push AS in a final response from a Related User is passed unmodified in the `UPDATE` message to the terminating entity.

When the target is a Related User and the served user has OIR, the served user has OIR settings to protect the served user’s identity, the Privacy header “history” is escaped within the `hi-targeted-to-uri` field containing the served user’s identity in the `History-Info` header in all messages containing `History-Info` headers which are passed to the terminating entity. This applies whether the `History-Info` header was included by the MTAS or was already in the received response. The terminating MTAS and CSCF are responsible for the interpretation of the Privacy header field.

When the target is the Primary User, OIR operates in the normal OIR mode.

Restriction of the served user’s identity is applied to all messages in the same direction.

Terminating Identification Restriction

Terminating Call Push

A `P-Asserted-Identity` header field received by the Call Push AS in a final response from a Primary or Related user is passed unmodified in the `UPDATE` message to the originating entity.

A `Privacy` header field received by the Call Push AS in a final response from a Related User is passed unmodified in the `UPDATE` message to the originating entity.



When the target is a Related User and the served user has TIR, the Privacy header “history” is escaped within the `hi-targeted-to-uri` field containing the served user’s identity in the `History-Info` header in all messages containing `History-Info` headers which are passed to the originating entity. This applies whether the `History-Info` header was included by the MTAS or was already in the received response. The originating MTAS and CSCF are responsible for the interpretation of the `Privacy` header field.

When the target is the Primary User, TIR operates in the normal TIR mode. Restriction of the served user’s identity is applied to all messages in the same direction.

Terminating Identity Presentation

Originating Call Push

When the target is a Primary User, the TIP service operates on all messages towards the originating entity.

2.3.18 Number Translation

Initial `INVITE` requests sent by the Call Push service to Related Users are subject to the Number Translation service.

2.3.19 Priority Call

Terminating Call Push

The MTAS passes on the contents of any `Priority` header received in the incoming initial `INVITE` in the outgoing `INVITE` to a Primary User.

The MTAS replaces any `Priority` header received in the incoming initial `INVITE` with a `Priority` header reflecting the priority of the served user in the outgoing `INVITE` requests to Call Push Related Users.

Originating Call Push

The MTAS adds any `Priority` header reflecting the priority of the served user in outgoing initial `INVITE` requests to Call Push Related Users.

2.3.20 Real-time Transfer of Tariff Information

After transfer of the session to a Related User, real-time Transfer of Tariff Information (RTTI) received from terminating network is sent to OCS for the originating charging session to Related User.



The received RTTI is removed from the SIP messages before passing them to the other side which remains in the communication.

2.3.21 Subscriber Credit Notification

Credit warning and credit limit reached announcements during a transferred session is played to the credit owner only if the user is involved in the transferred call. Those cases are handled as normal mid-session and session termination credit announcements.

After transfer of the session to a Primary User, the credit announcements are played for the Served User.

After transfer of the session to a Related User, no credit announcements are played to the Related User.

2.3.22 3PTY

Originating Call Push

An IMS user can use the Call Push service, as an originating 3PTY Session Participant and transfer a 3PTY session to a Primary or Related user. The “isfocus” feature parameter is forwarded to the Primary or Related user.

An IMS user cannot use the Call Push service, as an originating 3PTY Session Originator and transfer a 3PTY session. If Call Push is started, the 3PTY Session is terminated.

Terminating Call Push

An IMS user can use the Call Push service, as a terminating 3PTY Session Participant and transfer a 3PTY session to a Primary or Related user.

The “Isfocus” feature parameter is forwarded to the Primary or Related user.

An IMS user cannot use the Call Push service, as a terminating 3PTY Session Originator and transfer a 3PTY session. If Call Push is started, the 3PTY Session is terminated.

Call Pull

Call Pull is not available for 3PTY sessions in case the served user is the 3PTY initiator.



2.3.23 Video Fallback to Audio

Initial `INVITE` requests sent by the Call Push service to Related Users can be subject to the Video Fallback to Audio service.

2.3.24 OCT

Call Pull is available for sessions transferred by operator (directory lookup services).

3 Subscription Rules

The STOD subscription rules are expressed with XML defined, by a schema and follow the structure illustrated in Figure 1.

The STOD service has a single target-list and a single Rule set, and optionally a single divert-primary.

The STOD rule set consists of zero or many rules and one rule consists of zero or many conditions and one action.

The STOD service can also use targets from the common target list and from the common device list.

For more information about target handling for the STOD service, refer to *MTAS Target Handling Management Guide*.

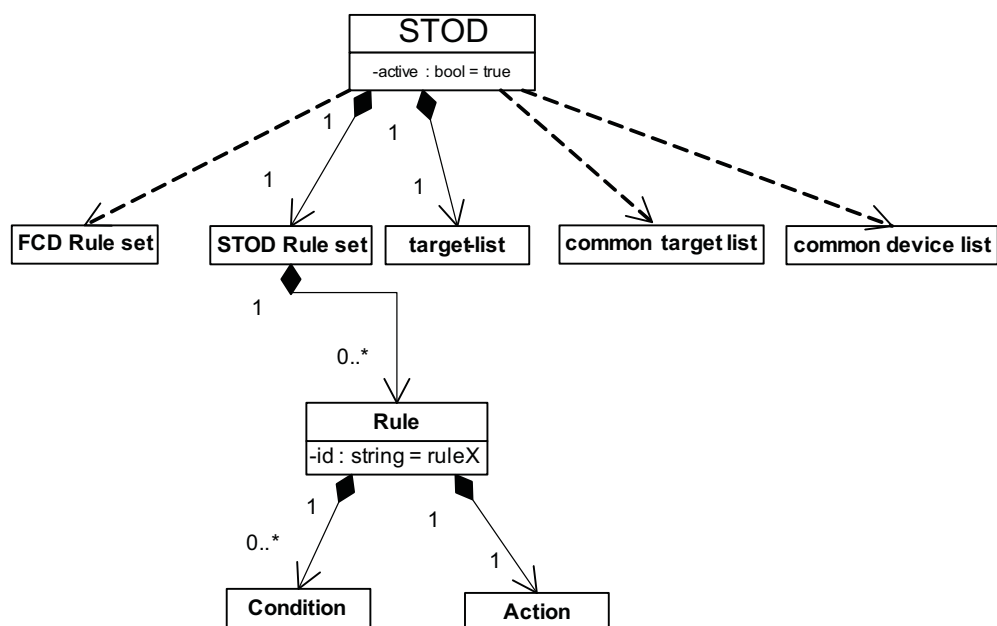


Figure 1 STOD Subscription Elements

When the STOD rule set consists of zero rules, it uses the rules from the FCD rule set, if exist.

The element `<target-list>` can contain up to 10 elements (`mmt-serv:target`) which contain the ID and name attributes that are the list of identities which can be included in the rule actions.

The element *Rule* (`<fcd-rule>`) has one *id* attribute that identifies the rule. It has nothing to do with the evaluation order.

The top-level STOD element has an attribute `active` of type Boolean. If `active=true`, then the rule set is evaluated.

The `<conditions>` element is a grouping element for conditions for a rule. All conditions must be satisfied for the rule to take effect. If no conditions are present, then the rule is always applicable.

3.1 Conditions

The different conditions that apply to STOD are as follows:

- `cp:validity`
All other common policy (cp) conditions are not used by the STOD. For more details and definitions, refer to [IETF RFC 4745](#).
- `ss:rule-deactivated`
- `mmt-serv:valid-periods`
- `mmt-serv:invalidity`

3.1.1 `cp:validity`

Specifies one of 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, a starting and 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. (2005-12-24 12.00 in Stockholm, Coordinated Universal Time (UTC) +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.

Note: When setting the value of these attributes, the Daylight Saving Time correction must be considered as well.

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 `cp:validity` conditions are converted to UTC before being compared to the current time, also in UTC. An example of this is shown in Example 1.

```
<cp:validity>
<!-- This example shows time being defined as an offset from UTC-->
  <cp:from>2008-10-12T20:00:00-08:00</cp:from>
  <cp:until>2008-10-19T19:59:59-08:00</cp:until>
  <!-- example in UTC as shown by the Z -->
  <cp:from>2008-11-27T20:00:00Z</cp:from>
  <cp:until>2008-11-28T08:00:00Z</cp:until>
</cp:validity>
```

Example 1 UTC Conversion

For further details on the configuration of time-based services in MTAS, refer to *MTAS Time Based Services Management Guide*.

3.1.2 ss: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.

3.1.3 mmt-serv:valid-periods

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

- `times of day` (in form of intervals, like 09:00-12:00)
- `days of the week` (for example, Monday)
- `workdays/non-workdays` (use of preprovisioned/configured list of weekdays)
- `private and public holidays` (for example, Monday)
- `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 element within the subconditions evaluates to TRUE, then the subcondition evaluates to TRUE. The `mmt-serv:valid-periods` evaluates to TRUE only when all the included subconditions evaluate to TRUE.

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

For more information on the other provisioned and configured data used at evaluation of `mmt-serv:valid-periods`, refer to *MTAS Time Based Services Management Guide*.

The details of the subconditions have been moved to *MTAS Time Based Services Management Guide*.

3.1.3.1 **mmt-serv:utc-offset**

The `mmt-serv:utc-offset` element is used to specify the time zone offset. The value of the offset can be + or - regarding UTC, for example, [+]. The value of an hour is 00-23, for example, [0-1][0-9]2[0-3]. For minutes, the value is 00-59, for example, [0-5][0-9].

If the `mmt-serv:utc-offset` element is omitted, the `mmt-serv:valid-periods` condition uses UTC. An example of a time zone offset is shown in Example 2.

```
<mmt-serv:utc-offset>+06:30</mmt-serv:utc-offset>
```

Example 2 Time Zone Offset

3.1.3.2 **mmt-serv:valid-days**

The `mmt-serv:valid-days` element is a subcondition. It is used to define days of the week on which the `mmt-serv:valid-periods` condition is evaluated to TRUE. One day or multiple days can be defined (Monday to Sunday). If `mmt-serv:valid-days` is omitted, the `mmt-serv:valid-periods` condition applies to all days of the week. An example of a valid-days subcondition is shown in Example 3.

```
<mmt-serv:valid-days>
  <mmt-serv:day>Monday</mmt-serv:day>
  <mmt-serv:day>Tuesday</mmt-serv:day>
  <mmt-serv:day>Wednesday</mmt-serv:day>
  <mmt-serv:day>Thursday</mmt-serv:day>
  <mmt-serv:day>Friday</mmt-serv:day>
</mmt-serv:valid-days>
```

Example 3 Valid-days Subcondition



3.1.3.3 mmt-serv:valid-times

The `mmt-serv:valid-times` element is a subcondition. It is used to define the intervals of the day within the condition is evaluated to TRUE. If `mmt-serv:valid-times` is omitted, the `mmt-serv:valid-periods` condition applies to all times of day, that is, 00:00-24:00. An example of a `valid-times` subcondition is shown in Example 4.

```
<mmt-serv:valid-times>
  <mmt-serv:interval from="09:00" until="12:00"/>
  <mmt-serv:interval from="13:00" until="17:00"/>
</mmt-serv:valid-times>
```

Example 4 Valid-times Subcondition

3.1.4 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 in Stockholm, UTC+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.

Note: When setting the value of these attributes, the Daylight Saving Time correction must be considered as well.

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 further details on the configuration of time-based services in MTAS, refer to *MTAS Time Based Services Management Guide*.



3.2 Actions

The rule only has one action. The actions supported in the rules within the ruleset are the following:

- `mmt-serv:parallel-distribution`
- `mmt-serv:serial-distribution`

3.2.1 `mmt-serv:parallel-distribution`

The action `mmt-serv:parallel-distribution` is associated with an optional “ring-period” attribute specifying the maximum time period for which the targets must be left ringing in parallel without an answer. It also has up to 10 `<target>` elements with the “name” attribute as the unique key. It is a reference by name to a target identity to which the communication must be distributed. The name in the rule must have a matching name in the STOD target list or in the common target list, or in the common device list from the User Common Data, or it has to be the special value `PRIMARY`. At least one `<target>` element must be present on creation of a `<parallel-distribution>` element.

For more information about target handling for the STOD service, refer to *MTAS Target Handling Management Guide*.

3.2.2 `mmt-serv:serial-distribution`

The action `mmt-serv:serial-distribution` has up to 10 `<target>` elements with the “name” attribute as the unique key. It is a reference by name to a target identity to which the communication must be distributed. The name in the rule must have a matching name in the STOD target list or in the common target list, or in the common device list from the User Common Data, or it has to be the special value `PRIMARY`. At least one `<target>` element must be present on creation of a `<serial-distribution>` element. Each `<target>` element is associated with an optional “ring-period” attribute specifying the maximum time period for which the target must be left ringing without an answer.

For more information about target handling for the STOD service, refer to *MTAS Target Handling Management Guide*.

`mmt-serv:target:`

The ID value of the target inside target-list which must be a valid SIP or tel URI. The target inside the distribution actions is just a reference by the name attribute to the target in the target list.

The format of the rules is checked by the XDMS. For more information, refer to *MTAS CAI3G Interface*.



If the service is active, then STOD evaluates the rules. The rules are evaluated from top to bottom, that is, rule n precedes rule $n + 1$, where n is the order of appearance, not the rule ID.

A rule is matched if all conditions evaluate to true. This means that the evaluation of conditions stops when the first condition is false (lazy evaluation). Only the first matching rule and its specified action elements apply.





4 STOD Configuration

The STOD service is controlled by the *MtasStod* MO. Some STOD-related parameters used by the SSC invocation of Call Pull are placed in the *MtasSscStod* MO. An overview of the STOD MO structure is shown in Figure 2.

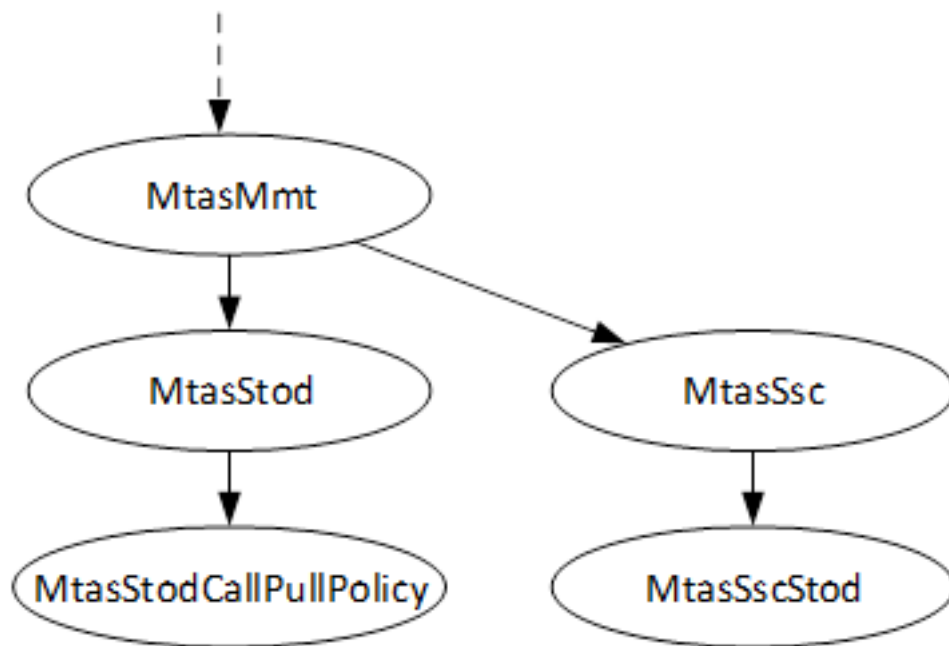


Figure 2 STOD MO Structure

For configurable MOs and attributes related to the STOD services, refer to *Managed Object Model (MOM)*.

4.1 STOD Administrative State Configuration

The STOD service is enabled by setting the `mtasStodAdministrativeState` attribute in the *MtasStod* MO to 1 (Unlocked). If the `mtasStodAdministrativeState` is set to 0 (Locked), no STOD service is provided by the MTAS.



4.2 Call Pull Configuration

4.2.1 Call Pull Service Code Syntax Configuration

For Call Pull service started through SSC, the operator must configure the syntax of the supplementary service code that starts the service. The configuration is done through the `mtasSscStodPullComSyntInv` attribute in the `mtasSscStod` MO.

4.2.2 Call Pull Policy Configuration

Call Pull policies are rules for allowing invocation of the Call Pull service by the users. Policies are based on evaluating the pulled session related or served user-related conditions.

Policies which are always active and cannot be controlled by the operator are described in Section 2.3 Interaction with Other Services on page 6. Policies which can be enabled and disabled by the operator, or where the reason text indicating the rejection of a call pull by a policy to the “pull initiating” user, are configured through the `MtasStodCallPullPolicy` MO.

For details of operator-controlled policies, refer to *Managed Object Model (MOM)*.

4.2.3 Call Pull Announcement Configuration

The Call Pull service contains configuration options for progress and Rejection announcements. Progress announcement is played to the “pull initiating” device during the call transfer. Rejection announcement is played to the “pull initiating” device when the call pull request is rejected due to a Call Pull policy or other unfulfilled pre-condition.

For Call Pull service started through SSC, configuration of the progress announcement is mandatory and set through the `mtasSscStodPullProgressAnnName` attribute in the `mtasSscStod` MO.

Playing a rejection announcement is mandatory and is controlled by setting the `mtasSscStodPullRejectAnnName` attribute in the `mtasSscStod` MO.

For Call Pull service started through `INVITE` with a `Replaces` header, configuration of the progress announcement is optional but highly recommended as precondition negotiation is only supported if progress announcement is configured. The announcement can be set through the `mtasStodPullProgressAnnName` attribute in the `mtasStod` MO. In case it is not set, no progress announcement is played during the call transfer is in progress. Playing a rejection announcement is optional and can be controlled by setting the `mtasStodPullRejectionAnnName` attribute in the `mtasStod` MO.



The announcement name used for setting the configuration attributes must be the name of an existing Generic Announcement previously configured in MTAS. For details for configuring Generic Announcements, refer to *MTAS Generic Announcement Management Guide*.

4.3 Wholesale for STOD Configuration

The STOD service supports Wholesale. STOD is configurable on Virtual Telephony Provider (VTP) level.

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

- The `vtasStodAdministrativeState` attribute in the *VtasStod* MO
- The `mtasStodAdministrativeState` attribute in the *MtasStod* MO

Configuration of the Call Pull Policy and Call Pull Announcement, described in previous sections, are also available on VTP level. Attributes corresponding to VTP level settings are found in the *VtasStodCallPullPolicy* MO and *VtasStod* MO, respectively.

For more information about the Wholesale service, refer to *MTAS Wholesale Support Management Guide*.

4.4 Session Timer Configuration

The `mtasStodHoldClearTimer` attribute controls after how many seconds the held session would be terminated when BYE is received instead of triggering the STOD Call Push service. The default value is 5 seconds.

If the `mtasStodHoldClearTimer` attribute is set to "0", the STOD Call Push service is suppressed for all users.

If the value "0" is set, the STOD Call Push service is suppressed on node level.

4.5 Service Data Configuration

This section describes how to configure the service data.

4.5.1 Operator Subscription Level Service Configuration

The operator can activate or deactivate the STOD service for the subscriber by providing the STOD XML structure using the CAI3G protocol to user subscriber data.

The operator can set or remove the following:

- List of destination targets
- Fixed-target attributes of the list of destination targets
- Maximum number of targets in the target-list
- STOD rules
- Maximum number of STOD rules

For more information about the XML structure, refer to *MTAS CAI3G Interface*.

4.5.2 Subscriber Subscription-Level Service Configuration

Subscriber subscription data is configured through the Ut interface.

The user can set or remove the following:

- target-list: A list of the related targets that can be included in communication distribution rules, for example, of the served user. Up to 10 target elements can be included in the target-list.

The following element exists:

- fixed-targets: When fixed-targets are set to `true` then the target identities are set by the operator and cannot be changed by the user.
- target: The target element allowing multiple instances with name as the unique key.

The following elements exist:

- name: The name is the key for the distribution target. This is the name by which STOD rules refer to targets. It must be present on the creation of a `<target>` element.
- id: The id is the identity of the target. It is a sip:URI or tel:URI. It must be present on the creation of a target.
- Rules: Define the rules.

The following elements exist:

- Conditions: which specify the validity of a rule.
- Actions: either serial, parallel, or flexible with up to 10 identities from either the STOD target-list, common target list, common device list, or PRIMARY (the default identifier for the Primary User).

Only targets defined in the common target list can be marked with the auto-answer avoidance feature.



For information on the XML schema for the Ut interface, refer to *MTAS Ut Structure*.

4.6 Reason-text Configuration in BYE Message

When the Call Pull mechanism is used to trigger transfer of a session from one (own) device to another, the initial call leg is terminated using BYE message. This BYE message can include the Reason header depending on the value of parameter `mtasStodPullByeReason` which defines the reason text for Reason header. The content of the attribute is copied to the reason text field. The value is to be added without quotation mark. If the value of `mtasStodPullByeReason` is set to `text Call has been transferred to another device`, then the Reason header in the BYE message looks like this:

```
Reason: SIP; cause=200; text="Call has been  
transferred to another device"
```

The protocol type and the cause code are non-configurable.

If the `mtasStodPullByeReason` attribute is empty, then the Reason header is not included in the BYE message.





5 Prerequisites for Full Operation of Call Pull Replaces

The following conditions must be fulfilled for Call Pull with replaces service to work satisfactory:

- For video Call Pull with replaces, all terminals involved in the Call Pull call must support Audio-Visual Profile Feedback (AVPF) with Full Intra-Request (FIR) and the terminals must also be able to handle SIP INVITEs with no SDP
- The media node must support all pulled media types
- Precondition media negotiation is only supported if call pull progress announcement is configured





6 Performance Management

For measurements related to the STOD service, refer to *Managed Object Model (MOM)*.





7 Fault Management

For alarms related to the STOD service, refer to *MTAS Alarm List*.





8 STOD Rule Examples

The examples in this section illustrate how the rules for STOD appear at the User Subscription Level.

8.1 Unconditional Parallel Ringing

This STOD rule executes parallel ringing to three devices, without any condition.

```
<cp:rule id="stod-unconditional">
  <cp:conditions>
    </cp:conditions>
  <cp:actions>
    <mmt-serv:parallel-distribution ring-period="60">
      <mmt-serv:target name="home"/>
      <mmt-serv:target name="car"/>
      <mmt-serv:target name="office"/>
    </mmt-serv:parallel-distribution>
  </cp:actions>
</cp:rule>
```

8.2 Time-based Parallel Ringing

A form of time mode can be achieved using the common policy validity condition. It uses absolute date and time values with offsets as required from UTC. If the current time falls between any of the from/until values, then the validity condition is satisfied, and the destination addresses are rung in parallel. For the parallel case, the ring period is applicable to all legs.



```
<cp:rule id="stod-time-mode-1">
  <cp:conditions>
    <cp:validity>
      <!-- This example shows time being defined as an offset from UTC-->
      <cp:from>2008-10-12T20:00:00-08:00</cp:from>
      <cp:until>2008-10-19T19:59:59-08:00</cp:until>

      <!-- example in UTC as shown by the Z -->

      <cp:from>2008-11-27T20:00:00Z</cp:from>
      <cp:until>2008-11-28T08:00:00Z</cp:until>
    </cp:validity>
  </cp:conditions>
  <cp:actions>
    <mmt-serv:parallel-distribution ring-period="60">
      <mmt-serv:target name="Alice"/>
      <mmt-serv:target name="Carol"/>
      <mmt-serv:target name="Bob"/>
    </mmt-serv:parallel-distribution>
  </cp:actions>
</cp:rule>
```

8.3 Serial Ringing Consecutive Days, Valid Periods

This STOD rule executes serial ringing with individual ring period defined for each leg for the valid periods defined for consecutive days.



```

<cp:rule id="STOD-time-mode-2-working-week">
  <cp:conditions>
    <mmt-serv:valid-periods>
      <mmt-serv:utc-offset>+05:30</mmt-serv:utc-offset>
      <mmt-serv:valid-days>
        <mmt-serv:day>Monday</mmt-serv:day>
        <mmt-serv:day>Tuesday</mmt-serv:day>
        <mmt-serv:day>Wednesday</mmt-serv:day>
        <mmt-serv:day>Thursday</mmt-serv:day>
        <mmt-serv:day>Friday</mmt-serv:day>
      </mmt-serv:valid-days>
      <mmt-serv:valid-times>
        <mmt-serv:interval from="09:00" until="12:00"/>
        <mmt-serv:interval from="13:00" until="17:00"/>
      </mmt-serv:valid-times>
    </mmt-serv:valid-periods>
  </cp:conditions>
  <cp:actions>
    <mmt-serv:serial-distribution>
      <mmt-serv:target name="Alice" ring-period="15"/>
      <mmt-serv:target name="PRIMARY" ring-period="20"/>
      <mmt-serv:target name="Carol" ring-period="30"/>
      <mmt-serv:target name="Dave" ring-period="35"/>
    </mmt-serv:serial-distribution>
  </cp:actions>
</cp:rule>

```