

DSLxChange, Adding New Media Type to SIP Stack

Broadcom Confidential

For a comprehensive list of changes to this document, see the [Revision History](#).

Broadcom, the pulse logo, Connecting everything, Avago, Avago Technologies, and the A logo are among the trademarks of Broadcom and/or its affiliates in the United States, certain other countries and/or the EU.

Copyright © 2017 by Broadcom. All Rights Reserved.

The term “Broadcom” refers to Broadcom Limited and/or its subsidiaries. For more information, please visit www.broadcom.com.

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Table of Contents

Introduction	4
Use Cases	4
Reception of INFO Requests	4
Reception of NOTIFY Requests.....	4
Reception of 200 OK Response for INVITE Request.....	5
Sending INFO Requests	5
Revision History	6

Broadcom Confidential

Introduction

This document describes how a user can add new media type support to the SIP stack. This feature has been added to release 5.02.L03. Older releases will require Broadcom to rebuild the SIP stack binary.

Note that the new media type will be accepted by the SIP stack; however, further handling will have to be implemented by the user. When the media type is accepted, the SIP stack passes an appropriate event to the application, and the media body is provided for further processing. The following use-cases show some of the ways the new media type may be handled.

Use Cases

The following are customer use cases where a customer may need a new media type supported. For illustrative purpose, this new media type is referred to as "application/test." In order to support this new media type, the customer must do the following:

- Add the new type to ESipMediaType and g_astSIPMEDIATYPE[].
- Call RegisterSupportedPayload(eMEDIATYPE_APPLICATION_NEW_MEDIA) in configSce().

The following are the required source code changes:

```
==== userspace/private/apps/voice/sig/cctk22/cfgsrc/cctkExtCfg.cpp ====
883a884
>   { "application", "test" }, // eMEDIATYPE_APPLICATION_TEST
==== userspace/private/apps/voice/sig/cctk22/src/cfg/cfgEngine.cpp ====
2724a2725
>   pSceCfg->RegisterSupportedPayload( eMEDIATYPE_APPLICATION_TEST );
==== userspace/private/apps/voice/sig/m5tua/CustomConfig/PreSceCoreCfg.h ====
200a201
>   eMEDIATYPE_APPLICATION_TEST, \
```

The following sections provide examples of how the new media type events can be handled.

Reception of INFO Requests

INFO requests are handled in CeMain::EvUnhandledRequest(). The handler processes the request, generates the eCEEVT_CALL_NET_INFO event, and sends that to the CCTK FSM handler. The following log would be seen when an INFO request with the new media type is received.

```
CeMain::EvUnhandledRequest - received 'application/test' media type
```

Reception of NOTIFY Requests

NOTIFY requests are handled in CeMain::EvResourceStateUpdate(). The handler processes the request, generates the eCEEVT_CALL_NET_NOTIFY event, and send that to the CCTK FSM handler. The following log would be seen when a NOTIFY request with the new media type is received.

```
CeMain::EvResourceStateUpdate - received 'application/test' media type
```

Reception of 200 OK Response for INVITE Request

The 200 OK responses to INVITE requests are handled in `CeMain::EvAnswered()`. The handler processes the request, generates the `eCEEVT_CALL_NET_ANSWER` event, and sends that to the CCTK FSM handler. The following log would be seen when a 200 OK response with the new media type is received.

```
CeMain::EvAnswered - received 'application/test' media type
```

Sending INFO Requests

INFO requests are completely user defined and the new media type can be used when configuring this SIP message. The following example is from `CeFsmCall::fsmDtmfInfoSnd()`. The user can replace `eMEDIATYPE_AUDIO_TELEPHONE_EVENT` with their new media type and modify the body appropriately.

```
MXD_GNS::CSipMessageBody *pBody =  
    BRCM_NEW( MXD_GNS::CSipMessageBody );  
...  
    pCtTypeHdr->GetContentTypeMType() =  
        MXD_GNS::GetMediaMType(MXD_GNS::eMEDIATYPE_AUDIO_TELEPHONE_EVENT).CStr();  
    pCtTypeHdr->GetContentTypeMSubType() =  
        MXD_GNS::GetMediaMSubType(MXD_GNS::eMEDIATYPE_AUDIO_TELEPHONE_EVENT).CStr();  
...  
    res = pBody->AddBody( pBlob, pCtTypeHdr );
```

Revision History

<i>Revision</i>	<i>Date</i>	<i>Change Description</i>
DSLxChange-AN1000-R	January 4, 2017	Initial release

Broadcom Confidential

Broadcom Confidential



Web: www.broadcom.com

Corporate Headquarters: San Jose, CA

© 2017 by Broadcom. All rights reserved.

DSLxChange-AN1000-R January 4, 2017