

**Digital Cinema Initiatives, LLC**

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

# **Digital Cinema System Specification**

## **Compliance Test Plan**

**Version 1.0**

**October 16, 2007**

**Copyright © 2007 by Digital Cinema Initiatives, LLC**

---

# Digital Cinema System Specification: Compliance Test Plan

## Important Notice:

This document is a Compliance Test Plan developed by Digital Cinema Initiatives, LLC (DCI). DCI is the owner of this Compliance Test Plan for the purpose of copyright and other laws in all countries throughout the world. The DCI copyright notice must be included in all reproductions, whether in whole or in part, and may not be deleted or attributed to others. DCI hereby grants to its members and their suppliers a limited license to reproduce this Compliance Test Plan for their own use, provided it is not sold. Others must obtain permission to reproduce this Compliance Test Plan from Digital Cinema Initiatives, LLC.

This Compliance Test Plan is intended solely as a guide for companies interested in developing products that can be compatible with other products developed using this document and the DCI Digital Cinema System Specification, Version 1.1. Each DCI member company shall decide independently the extent to which it will utilize, or require adherence to, this Compliance Test Plan. DCI shall not be liable for any exemplary, incidental, proximate or consequential damages or expenses arising from the use of this document. This document defines only one approach to compatibility, and other approaches may be available to the industry. Only DCI has the right and authority to revise or change the material contained in this document, and any revisions by any party other than DCI are unauthorized and prohibited.

Using this document may require the use of one or more features covered by proprietary rights (such as features which are the subject of a patent, patent application, copyright, mask work right or trade secret right). By publication of this document, no position is taken by DCI with respect to the validity or infringement of any patent or other proprietary right. DCI hereby expressly disclaims any liability for infringement of intellectual property rights of others by virtue of the use of this document. DCI has not and does not investigate any notices or allegations of infringement prompted by publication of any DCI document, nor does DCI undertake a duty to advise users or potential users of DCI documents of such notices or allegations. DCI hereby expressly advises all users or potential users of this document to investigate and analyze any potential infringement situation, seek the advice of intellectual property counsel, and, if indicated, obtain a license under any applicable intellectual property right or take the necessary steps to avoid infringement of any intellectual property right. DCI expressly disclaims any intent to promote infringement of any intellectual property right by virtue of the evolution or publication of this document.

---

DCI gratefully acknowledges the participation and technical contributions of CineCert LLC, 306 E. Alameda Avenue, Burbank, CA 91502 <http://www.cinecert.com/>, in the preparation of this document.

---

---

DCI gratefully acknowledges the participation and technical contributions of the Fraunhofer Institute for Integrated Circuits, IIS, Am Wolfsmantel 33, 91058 Erlangen, Germany, <http://www.iis.fraunhofer.de/>, in the preparation of this document.

---

# Table of Contents

1. Introduction .....	1
1.1. Overview .....	3
1.2. Audience .....	3
1.3. Conventions and Practices .....	3
1.3.1. Typographical Conventions .....	3
1.3.2. Documentation Format .....	4
1.4. Digital Cinema System Architecture .....	5
1.5. Strategies for Successful Testing .....	5
I. Procedural Tests .....	7
2. Digital Cinema Certificates .....	9
2.1. Certificate Structure .....	9
2.1.1. Basic Certificate Structure .....	11
2.1.2. SignatureAlgorithm Fields .....	12
2.1.3. SignatureValue Field .....	13
2.1.4. SerialNumber Field .....	14
2.1.5. SubjectPublicKeyInfo Field .....	15
2.1.6. RSA Key Format .....	16
2.1.7. Validity Field .....	17
2.1.8. AuthorityKeyIdentifier Field .....	18
2.1.9. KeyUsage Field .....	19
2.1.10. Basic Constraint Field .....	20
2.1.11. Public Key Thumbprint .....	21
2.1.12. Organization Name Field .....	23
2.1.13. OrganizationUnitName Field .....	24
2.1.14. Entity Name and Roles Field .....	25
2.1.15. Unrecognized Extensions .....	26
2.1.16. Signature Validation .....	27
2.1.17. Certificate Chains .....	28
2.2. Certificate Decoder Behavior .....	30
2.2.1. ASN.1 DER Encoding Check .....	30
2.2.2. Missing Required Fields .....	31
2.2.3. PathLen Check .....	33
2.2.4. OrganizationName Match Check .....	35
2.2.5. Certificate Role Check .....	36
2.2.6. Validity Date Check .....	37
2.2.7. Signature Algorithm Check .....	38
2.2.8. Public Key Type Check .....	39
2.2.9. Issuer Certificate Presence Check .....	40
3. Key Delivery Messages .....	41
3.1. eXtensible Markup Language .....	41
3.1.1. XML Documents .....	41
3.1.2. XML Schema .....	42
3.1.3. XML Signature Validation .....	43
3.2. Key Delivery Message Example .....	45
3.3. ETM features .....	50
3.3.1. ETM Structure .....	50
3.3.2. ETM Validity Date Check .....	51
3.3.3. ETM Signer Element .....	52
3.3.4. ETM EncryptionMethod Element .....	53
3.3.5. ETM AnnotationText Language .....	54
3.3.6. ETM ReferenceList Element .....	55

Version 1.0 ARCHIVED ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

3.3.7. ETM SignedInfo CanonicalizationMethod Element .....	56
3.3.8. ETM Signature Reference Elements .....	57
3.3.9. ETM SignatureMethod Element .....	58
3.3.10. ETM Signature Transforms Field .....	59
3.3.11. ETM Signature DigestMethod Element .....	60
3.3.12. ETM Signature Validity .....	61
3.4. KDM features .....	62
3.4.1. KDM MessageType Element .....	62
3.4.2. KDM SubjectName Element .....	63
3.4.3. KDM ContentAuthenticator Element .....	64
3.4.4. KDM Signer Certificate Presence .....	65
3.4.5. KDM KeyIdList/TypedKeyId Field .....	66
3.4.6. KDM ForensicMarkFlagList Element .....	67
3.4.7. KDM EncryptedData Element .....	68
3.4.8. KeyInfo Element .....	69
3.4.9. KDM DeviceListDescription Element .....	70
3.4.10. KDM ContentTitleText Language Attribute .....	71
3.4.11. KDM KeyType Scope Attribute .....	72
3.4.12. KDM EncryptionMethod .....	73
3.4.13. KDM CompositionPlaylistId Element .....	74
3.4.14. KDM Validity Fields .....	75
3.4.15. KDM KeyIdList Element .....	76
3.4.16. KDM CipherData Structure ID .....	77
3.4.17. KDM CipherData Signer Thumbprint .....	78
3.4.18. KDM CipherData Validity .....	79
3.4.19. KDM CipherData CPL ID .....	80
3.4.20. KDM EncryptedKey KeyType .....	81
3.4.21. KDM Recipient X509IssuerName .....	82
3.5. KDM Decoder Behavior .....	83
3.5.1. KDM NonCriticalExtensions Element .....	83
3.5.2. ETM IssueDate Field Check .....	84
3.5.3. Maximum Number of DCP Keys .....	85
3.5.4. Structure ID Check .....	86
3.5.5. Certificate Thumbprint Check .....	87
3.5.6. Certificate Presence Check .....	88
3.5.7. KeyInfo Field Check .....	89
4. Digital Cinema Packaging .....	91
4.1. Asset Map .....	91
4.1.1. Asset Map File .....	93
4.1.2. Volume Index File .....	94
4.2. Packing List .....	95
4.2.1. Packing List File .....	96
4.2.2. Packing List Signature Validation .....	98
4.3. Composition Playlist .....	99
4.3.1. Composition Playlist File .....	100
4.3.2. Composition Playlist Signature Validation .....	102
4.3.3. Track File Key Epoch .....	103
4.4. Track Files .....	104
4.4.1. MXF Internals .....	104
4.4.2. Image and Audio Packaging Standard .....	109
4.4.3. Timed Text Track File Format .....	111
4.4.4. Track File Length .....	113
4.4.5. Image Track File Frame Boundary .....	114

Version 1.0 ARCHIVED ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

- 4.4.6. Audio Track File Frame Boundary ..... 116
- 4.5. Essence ..... 118
  - 4.5.1. Image Structure Container and Image Container Format ..... 118
  - 4.5.2. Image Compression Standard & Encoding Parameters ..... 120
  - 4.5.3. Audio Characteristics ..... 122
  - 4.5.4. Timed Text Resource Encoding ..... 124
- 4.6. Digital Cinema Package ..... 126
  - 4.6.1. DCP Integrity ..... 126
- 5. Common Security Features ..... 129
  - 5.1. SPB Security Features ..... 129
    - 5.1.1. SPB Digital Certificate ..... 129
    - 5.1.2. SPB Type 2 Security Perimeter ..... 130
    - 5.1.3. SPB Type 2 Secure Silicon ..... 131
  - 5.2. Intra-Theater Communication ..... 132
    - 5.2.1. TLS Session Initiation ..... 132
    - 5.2.2. Auditorium Security Messages ..... 134
  - 5.3. Event Logs ..... 149
    - 5.3.1. Log Report Format ..... 149
    - 5.3.2. Event Log Operations ..... 153
    - 5.3.3. SM Proxy of Log Events ..... 158
  - 5.4. Security Log Events ..... 163
    - 5.4.1. Playout, Validation and Key Events ..... 163
    - 5.4.2. ASM and Operations Events ..... 170
- 6. Media Block ..... 181
  - 6.1. Security Manager (SM) ..... 181
    - 6.1.1. Image Integrity Checking ..... 181
    - 6.1.2. Sound Integrity Checking ..... 183
    - 6.1.3. Restriction of Keying to Monitored Link Decryptors ..... 184
    - 6.1.4. Restriction of Keying to MD Type ..... 185
    - 6.1.5. Restriction of Keying to valid CPLs ..... 186
    - 6.1.6. Remote SPB Integrity Monitoring ..... 187
    - 6.1.7. SPB Integrity Fault Consequences ..... 189
  - 6.2. Link Encryption (LE) ..... 190
    - 6.2.1. LDB Trust ..... 190
    - 6.2.2. Multiple LE Operation ..... 191
    - 6.2.3. LE Key Usage ..... 193
    - 6.2.4. IMB Link Encryption ..... 194
  - 6.3. Clocks and Time ..... 195
    - 6.3.1. Clock Adjustment ..... 195
    - 6.3.2. Clock Battery ..... 196
    - 6.3.3. Clock Resolution ..... 197
  - 6.4. Forensic Marking (FM) ..... 198
    - 6.4.1. FM Application Constraints ..... 198
    - 6.4.2. Granularity of FM Control ..... 199
    - 6.4.3. FM Payload ..... 200
  - 6.5. Image Reproduction ..... 201
    - 6.5.1. Playback of Image Only Material ..... 201
    - 6.5.2. Decoder Requirements ..... 202
  - 6.6. Audio Reproduction ..... 204
    - 6.6.1. Digital Audio Interfaces ..... 204
    - 6.6.2. Audio Sample Rate Conversion ..... 206
    - 6.6.3. Audio Delay Setup ..... 207
    - 6.6.4. Click Free Splicing of Audio Track Files ..... 209

Version 1.0 ARCHIVED ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

6.7. Timed Text Reproduction .....	210
6.7.1. Media Block Overlay .....	210
6.7.2. Timed Text Synchronization .....	211
6.7.3. Support for Multiple Captions .....	212
6.7.4. Default Timed Text Font .....	213
7. Projector .....	215
7.1. Projector Test Environment for Image Measurements .....	215
7.2. SPB Type 2 .....	216
7.2.1. Projector Physical Protection .....	216
7.2.2. Projector Access Door .....	217
7.2.3. SPB2 Requirements .....	218
7.2.4. SPB2 Secure Silicon Requirements .....	219
7.2.5. SPB2 Tamper Evidence .....	220
7.2.6. SPB2 Secure Silicon Field Replacement .....	221
7.2.7. Systems Without Electronic Marriage .....	222
7.2.8. Electronic Marriage Break Key Retaining .....	223
7.3. Companion SPB Type 1 .....	224
7.3.1. Projector Companion SPB Location .....	224
7.3.2. Companion SPBs with Electronic Marriage .....	225
7.3.3. Companion SPB Marriage Break Key Retaining .....	227
7.4. Link Decryptor Block .....	228
7.4.1. LDB without Electronic Marriage .....	228
7.4.2. LDB TLS Session Constraints .....	229
7.4.3. LDB Time-Awareness .....	230
7.4.4. LDB ASM Conformity .....	231
7.4.5. LDB Key Storage .....	232
7.4.6. LDB Key Purging .....	233
7.4.7. LDB Logging .....	235
7.5. Projector Image Reproduction .....	236
7.5.1. Projector Overlay .....	236
7.5.2. Projector Lens .....	237
7.5.3. Projector Pixel Count/Structure .....	238
7.5.4. Projector Spatial Resolution and Frame Rate Conversion .....	239
7.5.5. White Point Luminance and Uniformity .....	240
7.5.6. White Point Chromaticity and Uniformity .....	241
7.5.7. Sequential Contrast .....	242
7.5.8. Intra-frame Contrast .....	243
7.5.9. Grayscale Tracking .....	244
7.5.10. Contouring .....	245
7.5.11. Transfer Function .....	246
7.5.12. Color Accuracy .....	247
8. Screen Management System .....	249
8.1. Ingest and Storage .....	249
8.1.1. Storage System Ingest Interface .....	249
8.1.2. Storage System Capacity .....	250
8.1.3. Storage System Redundancy .....	251
8.1.4. Storage System Performance .....	252
8.2. Screen Management System .....	253
8.2.1. Screen Management System .....	253
8.2.2. Show Playlist Creation .....	254
8.2.3. Show Playlist Format .....	256
8.2.4. KDM Validity Checks .....	257
8.2.5. Automation Control and Interfaces .....	258

Version 1.6 ARCHIVED ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.7 released May 8, 2009

8.2.6. Interrupt Free Playback .....	259
8.2.7. Artifact Free Playback of Image Content .....	260
8.2.8. Restarting Playback .....	261
8.2.9. SMS User Accounts .....	262
8.2.10. SMS Operator Identification .....	263
8.2.11. SMS Identity and Certificate .....	264
8.2.12. Content Keys and TDL check .....	265
II. Design Evaluation Guidelines .....	267
9. FIPS Requirements for a Type 1 SPB .....	269
9.1. FIPS Testing Procedures .....	269
9.2. Submitted Materials .....	271
9.3. Test Lab Reports .....	272
9.4. Interpreting FIPS Test Reports .....	272
9.5. DCI Requirements for FIPS Modules .....	274
9.5.1. SM Operating Environment .....	274
9.5.2. LE Key Generation .....	274
9.5.3. SPB1 Tamper Responsiveness .....	274
9.5.4. Security Design Description Requirements .....	275
9.5.5. SPB1 Tamper Resistance .....	275
9.5.6. SPB1 FIPS Requirements .....	275
9.5.7. SPB1 Secure Silicon FIPS Requirements .....	275
9.5.8. Asymmetric Key Generation .....	276
9.5.9. Critical Security Parameter Protection .....	276
9.5.10. SPB 1 Firmware Modifications .....	276
10. DCI Requirements Review .....	279
10.1. Type 1 SPB Documentation .....	279
10.2. Type 2 SPB Documentation .....	280
10.3. Forensic Mark IP Disclosure .....	280
10.4. DCI Requirements for Security Modules .....	281
10.4.1. Theater System Reliability .....	281
10.4.2. Theater System Storage Security .....	281
10.4.3. Security Devices Self-Test Capabilities .....	281
10.4.4. Security Entity Physical Protection .....	281
10.4.5. Secure SMS-SM Communication .....	282
10.4.6. Location of Security Manager .....	282
10.4.7. SM Usage of OS Security Features .....	282
10.4.8. SM Secure Communications .....	282
10.4.9. Playback Preparation .....	283
10.4.10. SE Uniqueness Constraint .....	283
10.4.11. Prevention of Keying of Compromised SPBs .....	283
10.4.12. SPB Authentication .....	283
10.4.13. TLS Session Key Refreshes .....	283
10.4.14. LE Key Issuance .....	284
10.4.15. Maximum Key Validity Period .....	284
10.4.16. KDM Purge upon Expiry .....	284
10.4.17. Key Usage Time Window .....	284
10.4.18. Projector Secure Silicon Device .....	285
10.4.19. Access to Projector Image Signals .....	285
10.4.20. Systems with Electronic Marriage .....	285
10.4.21. Systems Without Electronic Marriage .....	285
10.4.22. Clock Date-Time-Range .....	285
10.4.23. Clock Setup .....	286
10.4.24. Clock Stability .....	286

Version 1.0 ARCHIVED ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

10.4.25. Repair and Renewal of SPBs .....	286
10.4.26. SPB2 Protected Devices .....	286
10.4.27. Clock Continuity .....	286
10.4.28. TLS Endpoints .....	287
10.4.29. Implementation of RRP's .....	287
10.4.30. SMS and SPB Authentication and ITM Transport Layer .....	287
10.4.31. Idempotency of ITM RRP's .....	287
10.4.32. RRP Synchronism .....	287
10.4.33. TLS Mode Bypass Prohibition .....	288
10.4.34. RRP Broadcast Prohibition .....	288
10.4.35. Implementation of Proprietary ITMs .....	288
10.4.36. RRP Initiator .....	288
10.4.37. SPB TLS Session Partners .....	288
10.4.38. SM TLS Session Partners .....	289
10.4.39. RRP "Busy" and unsupported Types .....	289
10.4.40. RRP Operational Message Ports .....	289
10.4.41. FM Generic Inserter Requirements .....	289
10.4.42. FM Algorithm General Requirements .....	290
10.4.43. FM Insertion Requirements .....	290
10.4.44. IFM Visual Transparency .....	290
10.4.45. IFM Robustness .....	290
10.4.46. AFM Inaudibility .....	291
10.4.47. AFM Robustness .....	291
10.4.48. FM Control Instance .....	291
10.4.49. SE Time Stamping .....	291
10.4.50. SE Log Authoring .....	291
10.4.51. SPB Log Storage Requirements .....	292
10.4.52. Remote SPB Log Storage Requirements .....	292
10.4.53. IMB Log Storage Capabilities .....	292
10.4.54. Logging for Standalone Systems .....	292
10.4.55. Logging of Failed Procedures .....	292
10.4.56. SPB Log Failure .....	293
10.4.57. Log Purging in failed SPBs .....	293
10.4.58. IMB Tasks .....	293
10.4.59. Private Keys outside Secure Silicon .....	293
10.4.60. Image Keys Outside Secure Silicon .....	293
10.4.61. Prohibition of SPB1 Field Serviceability .....	294
10.4.62. Use of Software Protection Methods .....	294
10.4.63. TMS role .....	294
10.4.64. D-Cinema Security Parameter Protection .....	294
10.4.65. RSA Key Entropy .....	294
10.4.66. Preloaded Symmetric Key Entropy .....	295
10.4.67. MD Caching of Keys .....	295
III. Consolidated Test Procedures .....	297
11. Testing Overview .....	299
11.1. Test Reports .....	299
12. Digital Cinema Package (DCP) Consolidated Test Sequence .....	301
12.1. Overview .....	301
12.2. DCP Test Sequence .....	301
13. Digital Cinema Server Consolidated Test Sequence .....	305
13.1. Overview .....	305
13.2. Server Test Sequence .....	305
14. Digital Cinema Projector Consolidated Test Sequence .....	313



Version 1.0 ARCHIVED ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

- 14.1. Overview ..... 313
- 14.2. Projector Test Sequence ..... 313
- 15. Digital Cinema Projector with IMB Consolidated Test Sequence ..... 319
  - 15.1. Overview ..... 319
  - 15.2. Projector Test Sequence ..... 319
- A. Test Materials ..... 329
  - A.1. Overview ..... 329
  - A.2. Images ..... 329
    - A.2.1. Introduction ..... 329
    - A.2.2. Sync Count ..... 329
    - A.2.3. Sync Count (Encrypted) ..... 330
    - A.2.4. 4K sync count ..... 330
    - A.2.5. Sync Count 48fps ..... 331
    - A.2.6. Channel I.D. 5.1 ..... 331
    - A.2.7. Channel I.D. 7.1 ..... 332
    - A.2.8. Channel I.D. 1-16 ..... 333
    - A.2.9. Pink Noise 7.1 ..... 334
    - A.2.10. 1 kHz Sine Wave ..... 335
    - A.2.11. 400 Hz Sine Wave ..... 336
    - A.2.12. "NIST" 2k Test Pattern ..... 336
    - A.2.13. "NIST" 4k Test Pattern ..... 337
    - A.2.14. Black to Gray Step Series ..... 338
    - A.2.15. 4K gray step ..... 338
    - A.2.16. Black to White Step Series ..... 339
    - A.2.17. 4K grayscale grad ..... 339
    - A.2.18. Color Accuracy Series ..... 340
    - A.2.19. Contouring ..... 341
    - A.2.20. Black (Empty Frame) ..... 341
    - A.2.21. White (White Frame) ..... 341
    - A.2.22. Checkerboard Frame ..... 342
    - A.2.23. Frame Number Burn-in ..... 342
    - A.2.24. Frame Number Burn-in, 48fps ..... 342
    - A.2.25. StEM 2K ..... 343
    - A.2.26. StEM 2K (Encrypted) ..... 343
    - A.2.27. StEM 4K ..... 343
    - A.2.28. StEM 4K (Encrypted) ..... 344
    - A.2.29. m01 pict foos ..... 344
    - A.2.30. m03 snd splc ..... 344
    - A.2.31. m09 pict bad hmac ..... 344
  - A.3. Sound ..... 345
    - A.3.1. Introduction ..... 345
    - A.3.2. Sync Count 5.1 ..... 345
    - A.3.3. Sync Count 5.1 (Encrypted) ..... 345
    - A.3.4. Sync Count 5.1 48fps ..... 346
    - A.3.5. Channel I.D. 5.1 ..... 346
    - A.3.6. Channel I.D. 7.1 ..... 347
    - A.3.7. Channel I.D. 1-16 ..... 347
    - A.3.8. Pink Noise, 16 Channels ..... 348
    - A.3.9. Pink Noise, 16 Channels, 96 kHz ..... 348
    - A.3.10. 1 kHz Sine Wave ..... 349
    - A.3.11. 1 kHz Sine Wave ..... 349
    - A.3.12. 400 hz sine wave ..... 349
    - A.3.13. Silence, 5.1 ..... 350

Version 1.0 ARCHIVED ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

A.3.14. Silence, 5.1 (Encrypted) ..... 350

A.3.15. StEM 5.1 Sound ..... 350

A.3.16. StEM 5.1 Sound (Encrypted) ..... 351

A.3.17. m02 snd foos ..... 351

A.3.18. m10 snd bad hmac ..... 351

A.4. D-Cinema Packages ..... 352

    A.4.1. Introduction ..... 352

    A.4.2. DCI 2K Sync Test ..... 352

    A.4.3. DCI 2K Sync test with Subtitles ..... 352

    A.4.4. DCI 2K Sync test with Subtitles (Encrypted) ..... 352

    A.4.5. DCI 2K Sync Test (48fps) ..... 353

    A.4.6. 4K Sync Test ..... 353

    A.4.7. DCI 5.1 Channel Identification ..... 353

    A.4.8. 5.1 Channel Identification ..... 353

    A.4.9. DCI 7.1 Channel Identification ..... 354

    A.4.10. 7.1 Channel Identification ..... 354

    A.4.11. DCI 0-16 Numbered Channel Identification ..... 354

    A.4.12. 0-16 Numbered Channel ..... 354

    A.4.13. DCI Gray Steps ..... 355

    A.4.14. DCI White Steps ..... 355

    A.4.15. DCI Grayscale Gradient ..... 355

    A.4.16. 4K Gray Steps ..... 355

    A.4.17. 4K Grayscale Gradient ..... 356

    A.4.18. Color Accuracy Series ..... 356

    A.4.19. Contouring Sequence ..... 356

    A.4.20. DCI NIST Frame with silence ..... 356

    A.4.21. DCI NIST Frame with Pink Noise ..... 357

    A.4.22. DCI NIST Frame with 1 kHz tone (-20 dB fs) ..... 357

    A.4.23. DCI NIST Frame with Pink Noise (96 kHz) ..... 357

    A.4.24. DCI NIST Frame with 1 kHz tone (-20 dB fs, 96kHz) ..... 357

    A.4.25. DCI NIST Frame no sound files ..... 358

    A.4.26. 4K NIST Pattern ..... 358

    A.4.27. DCI 2K Image with Frame Number Burn In ..... 358

    A.4.28. DCI 2K Image with Frame Number Burn In (48 fps) ..... 358

    A.4.29. DCI 2K Image with Frame Number Burn In (Flat) ..... 359

    A.4.30. DCI 2K Image with Frame Number Burn In (Scope) ..... 359

    A.4.31. DCI 2K StEM Test Sequence ..... 359

    A.4.32. DCI 2K StEM Test Sequence (Encrypted) ..... 359

    A.4.33. 4K StEM Test Sequence ..... 360

    A.4.34. 4K StEM Test Sequence (Encrypted) ..... 360

    A.4.35. 128 Reel Composition, "A" Series (Encrypted) ..... 360

    A.4.36. 128 Reel Composition, "B" Series (Encrypted) ..... 361

    A.4.37. DCI Black Spacer - 5 seconds ..... 361

    A.4.38. White Frame Sequence ..... 361

    A.4.39. Checkerboard Sequence ..... 361

    A.4.40. DCI Malformed Test 1: Picture with Frame-out-of-order error ..... 362

    A.4.41. DCI Malformed Test 2: Sound with Frame-out-of-order error ..... 362

    A.4.42. DCI Malformed Test 3: Sound Splice Tests ..... 362

    A.4.43. DCI DCP With an incorrect sound file (Wrong ID) ..... 362

    A.4.44. DCI DCP With an incorrect picture file (Wrong ID) ..... 363

    A.4.45. DCI CPL with incorrect trackfile hashes ..... 363

    A.4.46. DCI CPL with an Invalid Signature ..... 363

    A.4.47. DCI DCP with timed text and a missing font ..... 363

Version 1.0 ARCHIVED ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

A.4.48. DCI Malformed Test 9: Picture with HMAC error in MXF Track File ..... 364

A.4.49. DCI Malformed Test 10: Sound with HMAC error in MXF Track File ..... 364

A.4.50. DCI DCP 2K ..... 364

A.4.51. DCI DCP 2K, Malformed ..... 365

A.4.52. DCI DCP 4K ..... 365

A.4.53. Multi-line Subtitle Test ..... 365

A.4.54. Multi-line Subtitle Test w/o Integral Font ..... 366

A.5. Digital Certificates ..... 367

    A.5.1. Chain A1 IMB Certificate Files ..... 367

    A.5.2. Chain A2 IMB Certificate Files ..... 368

    A.5.3. Chain A3 IMB Certificate Files ..... 368

    A.5.4. Chain B1 Certificate Files ..... 378

    A.5.5. Chain C1 Certificate Files ..... 379

    A.5.6. Chain C3 Certificate Files ..... 379

A.6. Key Delivery Messages ..... 390

    A.6.1. Introduction ..... 390

    A.6.2. KDM with invalid XML ..... 390

    A.6.3. KDM that has expired ..... 390

    A.6.4. KDM with incorrect message digest ..... 390

    A.6.5. KDM with future validity period ..... 391

    A.6.6. KDM with empty TDL ..... 391

    A.6.7. KDM with imminent expiration date ..... 391

    A.6.8. KDM with no Forensic Marking enabled ..... 391

    A.6.9. KDM with Image Forensic Marking enabled ..... 392

    A.6.10. KDM with Audio Forensic Marking enabled ..... 392

    A.6.11. KDM with corrupted CipherData block ..... 392

    A.6.12. KDM with incorrect signer thumbprint ..... 393

    A.6.13. KDM without signer certificate ..... 393

    A.6.14. KDM without AuthorityKey certificate ..... 393

    A.6.15. KDM with KeyInfo mismatch ..... 394

    A.6.16. KDM with mismatched CipherData CPL ID ..... 394

    A.6.17. KDM without MessageType ..... 394

    A.6.18. KDM with invalid MessageType ..... 395

    A.6.19. KDM with expired Signer certificate ..... 395

    A.6.20. KDM issued before certificate valid ..... 395

    A.6.21. KDM validity exceeds signer validity ..... 396

    A.6.22. KDM without CarriedKeyName ..... 396

    A.6.23. KDM without EncryptedData ..... 396

    A.6.24. KDM with invalid message digest ..... 396

    A.6.25. KDM with mismatched keytype ..... 397

    A.6.26. KDM for multiple LDs, 2 LDBs ..... 397

    A.6.27. KDM for multiple LDs, 1 LD/LE, 1 LDB ..... 397

    A.6.28. KDM for multiple LDs, 2 LD/LE, 2 LDB ..... 398

    A.6.29. KDM for multiple LDs, 2 LD/LE, 1 LDB ..... 398

    A.6.30. KDM for 2K StEM ..... 398

    A.6.31. Expired KDM for 2K StEM ..... 399

    A.6.32. Image FM only KDM for 2K StEM ..... 399

    A.6.33. No FM KDM for 2K StEM ..... 399

    A.6.34. Sound Only FM KDM for 2K StEM ..... 399

    A.6.35. KDM for 128 Reel Composition, "A" Series ..... 400

    A.6.36. KDM for 128 Reel Composition, "B" Series ..... 400

    A.6.37. FM Constraints ..... 400

    A.6.38. KDM with non-empty NonCriticalExtensions ..... 400

Version 1.0 ARCHIVED ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8, 2009

B. Equipment List ..... 401  
    B.1. Hardware ..... 401  
    B.2. Software ..... 402  
C. Source Code ..... 405  
    C.1. Overview ..... 405  
    C.2. dc-thumbprint ..... 406  
        C.2.1. dc-thumbprint Source Code Listing ..... 406  
    C.3. schema-check ..... 408  
        C.3.1. schema-check Source Code Listing ..... 408  
    C.4. kdm-decrypt ..... 411  
        C.4.1. kdm-decrypt Source Code Listing ..... 411  
    C.5. j2c-scan ..... 416  
        C.5.1. j2c-scan Source Code Listing ..... 416  
    C.6. Eab\_calc.py ..... 420  
        C.6.1. Eab\_calc.py Source Code Listing ..... 420  
    C.7. uuid\_check.py ..... 422  
        C.7.1. uuid\_check Source Code Listing ..... 422  
    C.8. dsig\_cert.py ..... 424  
        C.8.1. dsig\_cert.py Source Code Listing ..... 424  
    C.9. dsig\_extract.py ..... 427  
        C.9.1. dsig\_extract.py Source Code Listing ..... 427  
D. ASM Simulator ..... 429  
E. GPIO Test Fixture ..... 435  
F. Reference Documents ..... 437  
G. DCI Specification v1.1 References to CTP ..... 441  
H. Abbreviations ..... 451  
Index ..... 453

Version 1.0 ARCHIVED ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 2009

---

# List of Figures

1.1. Typical DCI Compliant System Configuration .....	5
6.1. Audio Delay Timing .....	207
A.1. Sync Count .....	330
A.2. Channel I.D. 5.1 .....	332
A.3. Channel I.D. 1-16 .....	333
A.4. Pink Noise 7.1 .....	334
A.5. 1 kHz Sine Wave .....	335
A.6. "NIST" 2k Test Pattern .....	337
A.7. Black to Gray Step Series .....	338
A.8. Black to White Step Series .....	339
A.9. Color Accuracy Series .....	340
E.1. GPIO Test Fixture Schematic .....	435
E.2. GPIO Test Fixture Connector .....	435

**Page Intentionally Left Blank**

# List of Tables

4.1. Essence Container UL Values for d-cinema .....	105
4.2. Audio Samples Per Frame .....	116
4.3. Image Structure Operational Levels .....	119
6.1. ASM responder parameters .....	191
11.1. Test Session Data .....	299
12.1. Asset Map Procedures .....	301
12.2. Packing List Procedures .....	302
12.3. Composition Playlist Procedures .....	302
12.4. Track File Procedures .....	302
12.5. Image Essence Procedures .....	303
12.6. Sound Essence Procedures .....	303
13.1. Security Manager Certificate .....	305
13.2. Screen Manager Certificate .....	306
13.3. Power .....	306
13.4. Operator Roles .....	306
13.5. Screen Management System .....	306
13.6. KDM Ingest .....	307
13.7. Interface .....	308
13.8. Log Reporting .....	309
13.9. Security Events .....	310
13.10. Essence Reproduction .....	311
13.11. Media Block Security .....	311
13.12. Forensic Marking .....	312
14.1. Projector Certificate .....	313
14.2. Link Decryptor Certificate .....	314
14.3. Power .....	314
14.4. Secure Processing Block Type 2 .....	314
14.5. Interface .....	315
14.6. Security Events .....	316
14.7. Link Decryptor .....	316
14.8. Image Processing .....	317
15.1. Security Manager Certificate .....	319
15.2. Screen Manager Certificate .....	320
15.3. Projector Certificate .....	320
15.4. Power .....	321
15.5. Operator Roles .....	321
15.6. Screen Management System .....	321
15.7. KDM Ingest .....	322
15.8. Interface .....	323
15.9. Log Reporting .....	323
15.10. Security Events .....	323
15.11. Essence Reproduction .....	324
15.12. Media Block Security .....	325
15.13. Forensic Marking .....	326
15.14. Secure Processing Block Type 2 .....	326
15.15. Image Processing .....	327

**Page Intentionally Left Blank**



# List of Examples

2.1. D-Cinema Certificate .....	9
3.1. Packing List Example (Partial) .....	41
3.2. checksig execution .....	43
3.3. dsig-cert.py execution .....	43
3.4. An X.509 certificate in PEM format .....	44
3.5. dsig-extract.py execution .....	44
3.6. KDM - AuthenticatedPublic area .....	45
3.7. KDM - AuthenticatedPrivate area .....	46
3.8. KDM - Signature area .....	47
3.9. kdm-decrypt Usage and Output .....	49
4.1. Asset Map .....	91
4.2. Volume Index .....	92
4.3. Packing List .....	95
4.4. Composition Playlist .....	99
4.5. MXF Partition Header .....	104
4.6. Source Package structure .....	105
4.7. Cryptographic Framework and Cryptographic Context .....	106
4.8. Essence Descriptor for JPEG 2000 .....	106
4.9. Essence Descriptor for PCM Audio .....	107
4.10. MXF Random Index Pack (RIP) .....	108
5.1. Log Report Preamble Example .....	149
5.2. Log Report Record Example .....	150
5.3. Log Report Signature Example .....	151
C.1. dc-thumbprint execution .....	406
C.2. Using schema-check to check well-formedness .....	408
C.3. Using schema-check to check validity .....	408
C.4. kdm-decrypt execution .....	411
C.5. j2c-scan execution .....	416
C.6. Eab_calc.py execution .....	420
C.7. uuid_check.py execution .....	422
C.8. dsig_cert.py execution .....	424
C.9. dsig_extract.py execution .....	427

**Page Intentionally Left Blank**

---

# Chapter 1. Introduction

Digital Cinema Initiatives, LLC (DCI) is a joint venture of Disney, Fox, Paramount, Sony Pictures Entertainment, Universal, and Warner Bros. Studios. The primary purpose of DCI is to establish uniform specifications for d-cinema. These DCI member companies believe that d-cinema will provide real benefits to theater audiences, theater owners, filmmakers and distributors. DCI was created with the recognition that these benefits could not be fully realized without industry-wide specifications. All parties involved in d-cinema must be confident that their products and services are interoperable and compatible with the products and services of all industry participants. The DCI member companies further believe that d-cinema exhibition will significantly improve the movie-going experience for the public.

Digital cinema is today being used world-wide to show feature motion pictures to thousands of audiences daily, at a level of quality commensurate with (or better than) that of 35mm film release prints. Many of these systems are informed by the first *Digital Cinema System Specification, Version 1.0*, published by DCI in 2005. In areas of image and sound encoding, transport security and network services, today's systems offer practical interoperability and an excellent movie-going experience. These systems were designed, however, using de-facto industry practices.

With the publication of DCI's *Digital Cinema System Specification, Version 1.1* and the publication of required standards from SMPTE, ISO, and other bodies, it is becoming possible to design and build d-cinema equipment that meets all DCI requirements. Manufacturers preparing new designs, and theaters planning expensive upgrades are both grappling with the same question: how to know if a d-cinema system is *compliant* with DCI requirements?

## 1.1. Overview

This Compliance Test Plan (CTP) was developed by DCI to provide uniform testing procedures for d-cinema equipment. The CTP details testing procedures, reference files, design evaluation methods and directed test sequences for content packages and specific types of equipment. These instructions will guide the Test Operator through the testing process and the creation of a standard DCI compliance evaluation report.

This document is presented in three parts and six appendices.

- Part I: Procedural Tests — contains a library of test procedures for elements of a d-cinema system. Many of the test procedures are applicable to more than one element. The procedure library will be used in Part III to produce complete sequences for testing content and specific types of systems.
  - Chapter 2: *Digital Cinema Certificates* — describes test objectives and procedures to test d-cinema certificates and devices which use d-cinema certificates for security operations.
  - Chapter 3: *Key Delivery Messages* — describes test objectives and procedures to test Key Delivery Messages (KDM) and devices which decrypt KDM payloads.
  - Chapter 4: *Digital Cinema Packaging* — describes test objectives and procedures to test the files in a Digital Cinema Package (DCP).
  - Chapter 5: *Common Security Features* — describes test objectives and procedures to test security requirements that apply to more than one type of d-cinema device (*e.g.*, an SMS or a projector). Security event logging is also addressed in this chapter.
  - Chapter 6: *Media Block* — describes test objectives and procedures to test that Media Block device operations are correct and valid.
  - Chapter 7: *Projector* — describes test objectives and procedures to test that projector operations are correct and valid.

- Chapter 8: *Screen Management System* — describes test objectives and procedures to test that Screen Management System (SMS) operations are correct and valid.
- Part II: Design Evaluation Guidelines, contains two chapters that describe DCI security requirements for the design and implementation of d-cinema equipment, and methods for verifying those requirements through document analysis. Requirements in this part of the CTP cannot easily be tested by normal system operation. [FIPS-140-2] requirements for deriving random numbers, for example, must be verified by examining the documentation that is the basis of the FIPS certification.
- Chapter 9: *FIPS Requirements for a Type 1 SPB* — provides a methodology for evaluating the results of a FIPS 140-2 security test. Material submitted for testing and the resulting reports are examined for compliance with [DCI-DCSS-1-1] requirements.
- Chapter 10: *DCI Requirements Review* — provides a methodology for evaluating system documentation to determine whether system aspects that cannot be tested by direct procedural method are compliant with [DCI-DCSS-1-1] requirements.
- Part III: Consolidated Test Procedures contains consolidated test sequences for testing d-cinema equipment and content.
  - Chapter 11: *Testing Overview* — Provides an overview of the consolidated testing and test reports and a standard form for reporting details of the testing environment.
  - Chapter 12: *Digital Cinema Package (DCP) Consolidated Test Sequence* — A directed test sequence for testing a Digital Cinema Package (DCP).
  - Chapter 13: *Digital Cinema Server Consolidated Test Sequence* — A directed test sequence for testing a stand-alone Digital Cinema Server comprising an Image Media Block (IMB) and a Screen Management Server (SMS).
  - Chapter 14: *Digital Cinema Projector Consolidated Test Sequence* — A directed test sequence for testing a stand-alone Digital Cinema Projector with Link Decryptor Block (LDB).
  - Chapter 15: *Digital Cinema Projector with IMB Consolidated Test Sequence* — A directed test sequence for testing a Digital Cinema Projector having an integrated IMB and an integrated or external SMS.
- Appendix A: *Test Materials* — Provides a complete description of all reference files used in the test procedures including Digital Cinema Packages, KDMs and Certificates.
- Appendix B: *Equipment List* — Provides a list of test equipment and software used to perform the test procedures. The list is not exclusive and in fact contains many generic entries intended to allow Testing Organizations to exercise some discretion in selecting their tools.
- Appendix C: *Source Code* — Provides computer programs in source code form. These programs are included here because suitable alternatives were not available at the time this document was prepared.
- Appendix D: *ASM Simulator* — Provides documentation on **asm-requester** and **asm-responder**, two programs that simulate the behavior of devices that send and receive Auditorium Security Messages.
- Appendix E: *GPIO Test Fixture* — Provides a schematic for a GPIO test fixture.
- Appendix F: *Reference Documents* — Provides a complete list of the documents referenced by the test procedures and design requirements.
- Appendix G: *DCI Specification v1.1 References to CTP* — Provides a cross reference of [DCI-DCSS-1-1] sections to the respective CTP sections.
- Appendix H: *Abbreviations* — Provides explanations of the abbreviations used in this document.

## 1.2. Audience

This document is written to inform readers from many parts of the motion picture industry, including manufacturers, content producers, distributors and exhibitors. Readers will have specific needs from this text and the following descriptions will help identify the parts that will be most useful to them. Generally though, the reader should have technical experience with d-cinema systems and access to the required specifications. Some experience with general operating system concepts and installation of source code software will be required to run many of the procedures.

### Equipment Manufacturers

To successfully pass a compliance test, manufacturers must be aware of all requirements and test procedures. In addition to understanding the relevant test sequence and being prepared to provide the Test Operator with information needed to complete the tests in the sequence, the manufacturer is also responsible for preparing the documentation called for in Part II.

### Testing Organizations and Test Operators

The Testing Organizations and Test Operators are responsible for assembling a complete test laboratory with all required tools and for guiding the manufacturer through the process of compliance testing. Like the manufacturer, Testing Organizations and Test Operators must be aware of all requirements and test procedures at a very high level of detail.

### System Integrators

Integrators will need to understand the reports issued by Testing Organizations. Comparing systems using reported results will be more accurate if the analyst understands the manner in which individual measurements are made.

## 1.3. Conventions and Practices

### 1.3.1. Typographical Conventions

This document uses the following typographical conventions to convey information in its proper context.

A **Bold Face** style is used to display the names of commands to be run on a computer system.

A `Fixed Width` font is used to express literal data such as string values or element names for XML documents, or command-line arguments and output.

Examples that illustrate command input and output are displayed in a `Fixed Width` font on a shaded background:

```
$ echo "hello, world"  
hello, world
```

Less-than (<) and greater-than (>) symbols are used to illustrate generalized input values in command-line examples. They are placed around the generalized input value, *e.g.*, <input-value>. These symbols are also used to direct command output in some command-line examples, and are also an integral part of the XML file format.

Callouts (I) are used to provide reference points for examples that include explanations. Examples with callouts are followed by a list of descriptions explaining each callout.

Square brackets ([ and ]) are used to denote an external document reference, *e.g.*, [SMPTE-377M].

## 1.3.2. Documentation Format

The test procedures documented in Part I will contain the following sub-sections (except as noted).

### **Objective** —

Describes what requirements or assertions are to be proven by the test.

### **Procedures** —

Defines the steps to be taken to prove the requirements or assertions given in the corresponding objective.

### **Material** —

Describes the material (reference files) needed to execute the test. This section may not be present, for example, when the objective can be achieved without reference files.

### **Equipment** —

Describes what physical equipment and/or computer programs are needed for executing the test. The equipment list in each procedure is assumed to contain the Test Subject. If the equipment list contains one or more computer programs, the list is also assumed to contain a general purpose computer with a POSIX-like operating system (*e.g.*, Linux). This section may not be present, for example, when the objective can be achieved by observation alone.

### **References** —

The set of normative documents that define the requirements or assertions given in the corresponding objective.

### **The following language is used to identify persons and organizations by role:**

#### *Testing Organization*

An organization which offers testing services based on this document.

#### *Test Operator*

A member of the Testing Organization that performs testing services.

#### *Testing Subject*

A device or computer file which is the subject of a test based on this document.

### **The following language is used for referring to individual components of the system or the system as a whole:**

#### *Image Media Block and Controlling Devices*

This term refers to the combination of Image Media Block (IMB), Screen Management System (SMS) or Theater Management System (TMS), content storage and all cabling necessary to interconnect these devices. Depending upon actual system configuration, all of these components may exist in a single chassis or may exist in separate chassis. Some or all components may be integrated into the projector (see below).

#### *Projector*

The projector is the device responsible for converting the electrical signals from the Media Block to a human visible picture on screen. This includes all necessary power supplies and cabling.

#### *Projection System*

A complete exhibition system to perform playback of d-cinema content. This includes all cabling, power supplies, content storage devices, controlling terminals, media blocks, projection devices and sound processing devices necessary for a faithful presentation of the content.

#### *Theater System*

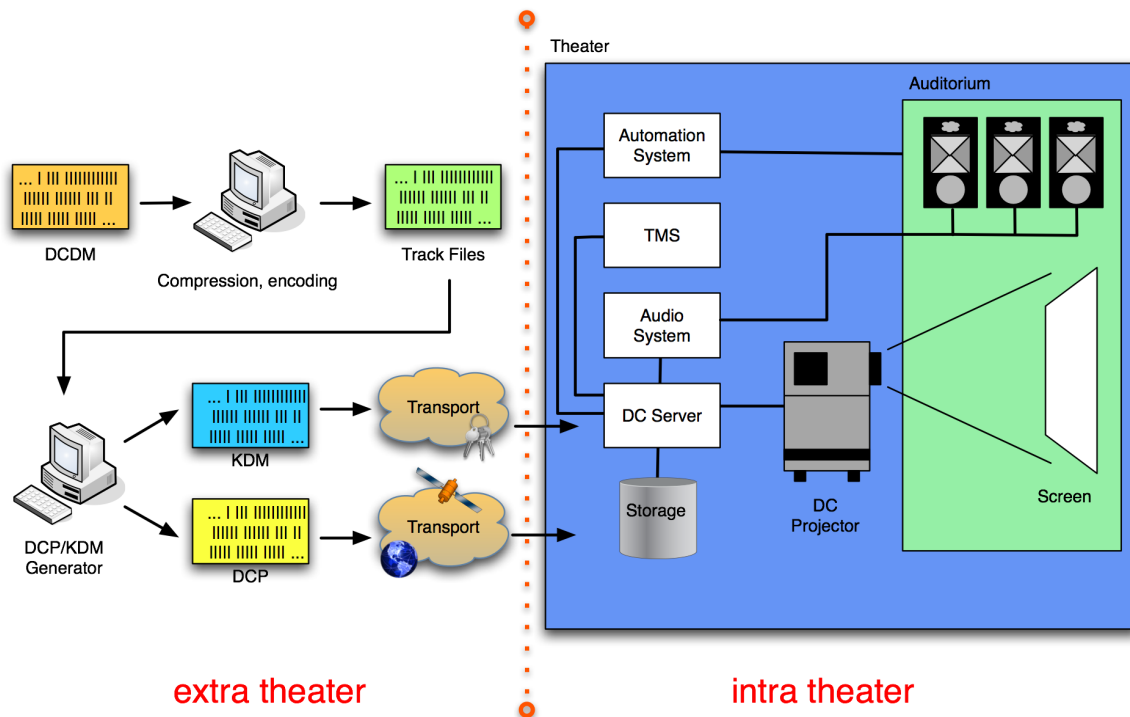
The projection system plus all the surrounding devices needed for full theater operations including theater loudspeakers and electronics (the "B-Chain"), theater automation, a theater network, and management workstations (depending upon implementation), etc.

Note: there may be additional restrictions, depending on implementation. For example, some Image Media Blocks may refuse to perform even the most basic operations as long as they are not attached to an SMS or Projector. For these environments, additional equipment may be required.

## 1.4. Digital Cinema System Architecture

The [DCI-DCSS-1-1] allows different system configurations, meaning different ways of grouping functional modules and equipment together. The following diagram shows what is considered to be a typical configuration allowed by DCI.

**Figure 1.1. Typical DCI Compliant System Configuration**



The left side of the diagram shows the *extra-theater* part, which deals with DCP and KDM generation and transport. The right side shows the *intra-theater* part, which shows the individual components of the projection system and how they work together. This test plan will test for proper DCP and KDM formats (*i.e.*, conforming to the DCI Specification), for proper transport of the data and for proper processing of valid and malformed DCPs and KDMs. In addition, physical system properties and performance will be tested in order to ensure that the system plays back the data as expected and implements all security measures as required by DCI.

While the above diagram shows what is considered to be a typical configuration allowed by the DCI Specification, the [DCI-DCSS-1-1] still leaves room for different implementations, for example, some manufacturers may choose to integrate the Media Decryptor blocks into the projector, or share storage between d-cinema servers.

## 1.5. Strategies for Successful Testing

In order to successfully execute one of the test sequences given in Part III, the Test Operator must understand the details of many documents and must have assembled the necessary tools and equipment to execute the tests. This document provides all the necessary references to standards, tutorials and tools to orient the technical reader.

As an example, Section 7.5.12 requires a calculation to be performed on a set of measured and reference values to determine whether a projector's colorimetry is within tolerance. Section C.6 provides an implementation of this calculation, but the math behind the program and the explanation behind the math are not presented in this document. The Test Operator and system designer must read the reference documents noted in Section 7.5.12 (and any references those documents may make) in order to fully understand the process and create an accurate design or present accurate results on a test report.

Preparing a Test Subject and the required documentation requires the same level of understanding as executing the test. Organizations may even choose to practice executing the test internally in preparation for a test by a Testing Organization.

The test procedures have been written to be independent of any proprietary tools. In some cases this policy has led to an inefficient procedure, but the resulting transparency provides a reference measurement that can be used to design new tools, and verify results obtained from any proprietary tools a Testing Organization may use.



---

# Part I. Procedural Tests

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

**Page Intentionally Left Blank**

---

---

# Chapter 2. Digital Cinema Certificates

Authentication of devices in D-Cinema is accomplished using *asymmetric cryptography*. Unlike symmetric cryptography, which uses the same key to encrypt and decrypt data, asymmetric cryptography uses a pair of keys that each reverse the other's cryptographic operations: data encrypted with one key in the key pair can only be decrypted by the other key in the key pair. In such a key pair, there is a *public key* that is distributed freely, and a *private key* that is closely held and protected. Public keys are not easily distinguished from one another because they don't carry any identifying information (they're just really long random numbers). To address this, public keys are distributed with metadata that describes the person or device that holds the private key, called the *subject*. This set of metadata and the public key comprise *digital certificate*. The standard that defines a digital certificate for Digital Cinema is [SMPTE-430-2]. It is based on the ITU standard for Public Key Infrastructure, called *X.509*, and specifies a number of constraints on the X.509v3 standard, such as the X.509 version that can be used and the size of the RSA keys, among other things.

A digital certificate also contains a *signature*, created by generating a message digest of the certificate and then encrypting that message digest with a (usually different) private key. The signature is then added to the certificate, and is used to verify that the certificate is authentic. The holder of the (private) key used to sign a certificate (encrypt the message digest) is known as the *issuer*, and identifying information about the issuer is in the Issuer field of the certificate, linking the issuer to the subject's certificate. Similarly, identifying information about the subject is in the Subject field. In most cases, the issuer and the subject are different. When the issuer and subject are the same, the certificate is known as being *self-signed*. A self-signed certificate is also self-validating, as its own public key is used to validate its signature. When a self-signed certificate is used to sign other certificates, it becomes the *Certificate Authority (CA)* for those certificates. The collection of certificates, from the top CA certificate to the last certificate (known as a *leaf certificate*) are collectively called the *certificate chain*.

Certificate authentication is recursive: in order to verify that a certificate is valid you have to decrypt the signature using the public key in the issuer's certificate. Once that signature is validated, if the issuer's certificate is not self signed then the signature validation process continues up the chain until a self-signed (CA) certificate is validated. A certificate is trusted only if its entire chain is valid.

The test procedures in this chapter are organized into two groups: tests that evaluate a certificate's compliance to [SMPTE 430-2] and tests that evaluate the behavior of devices that decode certificates. The Certificate Decoder tests are in this section because they are not specific to any particular type of system. All d-cinema devices that decode certificates must behave in the manner described by these tests.

## 2.1. Certificate Structure

The testing procedures that follow make use of the **OpenSSL** cryptographic tools and library. OpenSSL is a well known, free, and open source software package available for a number of hardware platforms and operating systems.

Much of the information in a digital certificate can be viewed in a human-readable format using OpenSSL's 'text' option. The information presented in the text output can be used to validate a number of certificate requirements, and to validate certificate-related KDM requirements by comparing the values present in the text output to the values in the KDM. The following example illustrates the features of a typical D-Cinema leaf certificate:

### Example 2.1. D-Cinema Certificate

```
$ openssl x509 -text -noout -in smpte-430-2-leaf-cert.pem 1
Certificate:
  Data:
    Version: 3 (0x2) 2
    Serial Number: 39142 (0x98e6) 3
    Signature Algorithm: sha256WithRSAEncryption 4
```

```

Issuer: O=.ca.example.com, OU=.ra-1b.ra-1a.s430-2.ca.example.com,
       CN=.cc-admin/dnQualifier=0sdCakNi3z6UPCYnogMFITbPMos= 5
Validity 6
  Not Before: Mar  9 23:29:52 2007 GMT 7
  Not After : Mar  8 23:29:45 2008 GMT 8
Subject: O=.ca.example.com, OU=.cc-admin.ra-1b.ra-1a.s430-2.ca.example.com, 9
       CN=SM.ws-1/dnQualifier=H/i8HyVmKEZSFoTeYI2UV9aBiq4=10
Subject Public Key Info: 11
  Public Key Algorithm: rsaEncryption 12
  RSA Public Key: (2048 bit) 13
    Modulus (2048 bit): 14
      [hexadecimal values omitted for brevity]
    Exponent: 65537 (0x10001) 15
X509v3 extensions: 16
  X509v3 Key Usage: 17
    Digital Signature, Key Encipherment, Data Encipherment 18
  X509v3 Basic Constraints: critical 19
    CA:FALSE
  X509v3 Subject Key Identifier: 20
    1F:F8:BC:1F:25:66:28:46:52:16:84:DE:60:8D:94:57:D6:81:8A:AE
  X509v3 Authority Key Identifier: 21
    keyid:D2:C7:42:6A:43:62:DF:3E:94:3C:26:27:A2:03:05:21:36:CF:32:8B
    DirName:/O=.ca.example.com/OU=.ra-1a.s430-2.ca.example.com/
           CN=.ra-1b/dnQualifier=3Nmh+Nx9WhnbDcXKK1puOjX41sY=
    serial:56:CE

Signature Algorithm: sha256WithRSAEncryption 22
  [hexadecimal values omitted for brevity]

```

## Certificate descriptions

- 1 Openssl command line and arguments to view the certificate text.
- 2 The x509 version of the certificate.
- 3 The serial number of the certificate.
- 4 The algorithm that was used to sign the certificate.
- 5 Information about the Issuer of the certificate.
- 6 The validity section of the certificate.
- 7 The date the certificate validity period begins.
- 8 The date the certificate validity period ends.
- 9 The Subject Name of the certificate.
- 10 Information about the Subject of the certificate.
- 11 Information about the Subject's public key.
- 12 The algorithm used to create the public key.
- 13 Information about the RSA public key.
- 14 One of the components of the used to calculate the public key.
- 15 One of the components of the used to calculate the public key.
- 16 x509 Version 3 Extensions. These extensions provide more information about the private key, the purposes for which it can be used, and how it is identified.
- 17 Key Usage. These are the actions that the private key can perform.
- 18 The enumerated list of actions that the private key can perform.
- 19 x509 Basic Constraints. These declare whether or not the certificate is a CA certificate, and whether or not there is a path length limitation. Basic Constraints must be marked Critical.
- 20 The Subject Key Identifier identifies the public key in the certificate.
- 21 The Authority Key Identifier identifies the Issuer key used to sign the certificate.
- 22 The Signature Algorithm used to sign the certificate.

## 2.1.1. Basic Certificate Structure

### Objective

Verify that the certificate uses the ITU X.509, Version 3 standard with ASN.1 DER encoding as described in [ITU-X509]. Also verify that the `Issuer` and `Subject` fields are present inside the signed part of the certificate.

### Procedures

The certificate format and encoding can be verified by using the `openssl` command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -inform PEM -in <certificate>
```

A correctly formatted and encoded certificate will be displayed as text output by `openssl`. An incorrectly formed certificate will cause `openssl` to display an error. A certificate that causes an error to be displayed by the `openssl` command is incorrectly formed and shall be cause to fail this test.

The version of the certificate and the presence of the `Issuer` and `Subject` fields in the signed portion of the certificate can be verified by viewing `openssl`'s text output of the certificate. The version number is indicated by **2** in the example certificate, and the issuer and subject fields are indicated by numbers **5** and **10**, respectively. An x509 version number other than 3, or the absence of either the `Subject` field or the `Issuer` field shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[ITU-X509]	
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.2. SignatureAlgorithm Fields

### Objective

Verify that the SignatureAlgorithm of the signature and the SignatureAlgorithm in the signed portion of the certificate both contain the value " sha256WithRSAEncryption".

### Procedures

The signature algorithms of the signature and of the certificate can be verified by using the **openssl** command to display the certificate text as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The signature algorithm of the certificate is indicated by **4** in the example certificate, and the signature algorithm of the signature is indicated by number **22** of the example certificate. Verify that these values are identical. Non-identical values shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.3. SignatureValue Field

### Objective

Verify that the `SignatureValue` field is present outside the signed part of the certificate and contains an ASN.1 Bit String that contains a PKCS #1 SHA256WithRSA signature block.

### Procedures

The certificate signature value can be verified by using the `openssl` command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

A correct certificate signature will be displayed as colon separated hexadecimal values in the text output by `openssl`. The signature block, omitted from the example certificate, will be present below the signature algorithm at the bottom of the output below callout number 22 of the example certificate. An incorrect certificate signature will cause `openssl` to display an error. A certificate that causes `openssl` to generate errors is cause to fail this test. A signature value other than `sha256WithRSAEncryption` is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.4. SerialNumber Field

### Objective

Verify that the `Serial Number` field is present inside the signed part of the certificate and that it contains a non-negative integer that is no longer than 64 bits (8 bytes).

### Procedures

The certificate serial number can be verified by using the `openssl` command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The serial number field is indicated by **S** in the example certificate. Confirm that the serial number is a non-negative integer that is no longer than 64 bits (8 bytes), and that the parenthetical phrase "neg" is not present. A negative serial number or a number larger than 64 bits shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl



## 2.1.5. SubjectPublicKeyInfo Field

### Objective

Verify that the Subject Public Key Info field is present inside the signed part of the certificate and that it describes an RSA public key with a modulus length of 2048 bits and a public exponent of 65537.

### Procedures

The subject public key info can be verified by using the **openssl** command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The Subject Public Key Info is indicated by **11** in the example certificate. The modulus length and the public exponent are indicated by **14** and **15**, respectively. Verify that the modulus length is 2048 bits and that the exponent is 65527. Any other value for the modulus length or the exponent shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.6. RSA Key Format

### Objective

Verify that RSA public key is 2048 bits in length.

### Procedures

The size of the public key can be verified by using the **openssl** command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The size of the public key is indicated in parentheses next to the Subject Public Key key section, which is indicated by callout **13** in the example certificate. Confirm that this value is "2048 bits". Any other value is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.7.6
[RFC-3447]	
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.7. Validity Field

### Objective

Verify that the `Validity` field is present inside the signed part of the certificate and contains timestamps in UTC. Timestamps with years up to and including 2049 must use two digits (`UTCTime`) to represent the year. Timestamps with the year 2050 or later must use four digits (`GeneralizedTime`) to represent the year.

### Procedures

The presence of the validity field can be verified by using the `openssl` command to display the certificate text as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The validity field is indicated by callout 6 in the example certificate. Confirm that the field is present and that it contains a "Not Before" value as a UTC timestamp as indicated by 7 of the example certificate and a "Not After" value as a UTC timestamp as indicated by 8 of the example certificate.

Verifying the format of the timestamps as either `UTCTime` or `GeneralizedTime` can be accomplished by viewing the ASN.1 sequences of the certificate with `OpenSSL`. Additionally, by using the `grep` command to specify a text string to display, in this case, "TIME", the time formats can be quickly identified:

```
$ openssl asn1parse -in <certificate> |grep TIME
154:d=3 hl=2 l= 13 prim: UTCTIME          :070312145212Z
169:d=3 hl=2 l= 13 prim: UTCTIME          :270307145212Z
```

Confirm that timestamps up to the year 2049 are in `UTCTime` format, and that timestamps starting with the year 2050 are in `GeneralizedTime` format. Timestamps in `UTCTime` format will be formatted as "YYMMDDhhmmssZ", and Timestamps in `GeneralizedTime` format will have the year coded as "YYYYMMDDhhmmssZ", where "Y" represents the year, "M" represents the month, "D" represents the day, and "h", "m", "s", and "Z" represent hours, minutes, seconds, and the Universal Coordinated Time zone. A timestamp prior to 2049 that is not in UTC format shall be cause to fail this test. A timestamp starting in 2050 or later that is not in `GeneralizedTime` format shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-2]	9.8

Test Equipment
openssl

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 2.1.8. AuthorityKeyIdentifier Field

### Objective

Verify that the Authority Key Identifier field is present in the X509v3 Extensions section inside the signed part of the certificate.

### Procedures

The presence of the Authority Key Identifier field can be verified by using the **openssl** command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The Authority Key Identifier of the certificate is indicated by **21** in the example certificate. Confirm that this field exists. The absence of the Authority Key Identifier field shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.9. KeyUsage Field

### Objective

Verify that the Key Usage field is present in the X509v3 Extensions section inside the signed part of the certificate. For signer certificates, verify that only the Certificate Sign flag is true, and for leaf certificates, Certificate Sign (KeyCertSign) is false or not present and that the Digital Signature and Key Encipherment flags are true.

### Procedures

The presence of the Key Usage field can be verified by using the **openssl** command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The Key Usage field in the certificate is indicated by **17** in the example certificate. For signing certificates, confirm that this field exists and that the only key usage listed in the usage list (indicated by **18**) is "Certificate Sign" (KeyCertSign). For leaf certificates, confirm that the key usages listed are "Digital Signature" and "Key Encipherment". Absence of the Key Usage field shall be cause to fail this test. For a signer certificate, the absence of the "Certificate Sign" flag or the presence of any other flag shall be cause to fail this test. For a leaf certificate, the presence of the "Certificate Sign" sign flag, or the absence of either the "Digital Signature" or "Key Encipherment" flags shall be cause to fail this test.

Note that leaf certificates may have other Key Usages specified, and the presence of other usages not specifically referenced here shall not be a reason to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.10. Basic Constraint Field

### Objective

Verify that the `Basic Constraints` field is present in the `X509v3 Extensions` section of the signed portion of the certificate. For signer certificates, verify that the certificate authority attribute is true (`CA:TRUE`) and the `PathLenConstraint` value is present and either zero or positive. For leaf certificates, verify that the certificate authority attribute is false (`CA:FALSE`) and the `PathLenConstraint` is absent or zero.

### Procedures

The presence of the `Basic Constraints` field can be verified by using the `openssl` command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The `Basic Constraints` field in the certificate is indicated by **19** in the example certificate. For signing certificates, confirm that this field exists, that the certificate authority value is true (`CA:TRUE`), and that the path length is present and is a positive integer. For leaf certificates, confirm that the certificate authority value is false (`CA:FALSE`) and that the path length is absent or zero. The absence of the `Basic Constraints` field shall be cause to fail this test. For signer certificates, the absence of the `CA:TRUE` value, or a negative or missing Path Length value shall be cause to fail this test. For leaf certificates, the presence of the `CA:TRUE` value or the presence of a path length greater than zero shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.11. Public Key Thumbprint

### Objective

Verify that there is exactly one `DnQualifier` present in the `Subject` field and that the `DnQualifier` value is the Base64 encoded thumbprint of the subject public key in the certificate. Also verify that there is exactly one `DnQualifier` present in the `Issuer` field and that the `DnQualifier` value is the Base64 encoded thumbprint of the issuer's public key.

### Procedures

The presence of a single instance of the `DnQualifier` field can be verified by using the `openssl` command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The `Subject DnQualifier` in the certificate is in the `Subject` information as indicated by **10** in the example certificate, and the `Issuer DnQualifier` in the certificate is in the `Issuer` information as indicated by **5**. Confirm that each of these fields contain only one `DnQualifier`. Missing `DnQualifier` values in either of these fields or the presence of more than one `DnQualifier` in either field shall be cause to fail this test.

The public key `DnQualifier` must be recalculated to confirm that the `DnQualifier` value in each of these fields is correct. The following steps perform this calculation:

1. Extract the public key from the certificate (using `OpenSSL`)
2. Convert the public key from Base64 to binary (using `OpenSSL`)
3. Skip 24 bytes into the binary form of the public key (using `dd`)
4. Calculate the SHA-1 digest over the remaining portion of the binary form of the public key (using `OpenSSL`)
5. Convert the SHA-1 digest value to Base64 (using `OpenSSL`)

The steps above can be performed in sequence by redirecting the output from one step to the next, and using `openssl` and the `dd` command present on most posix compliant operating systems, such as:

```
$ openssl x509 -pubkey -noout -in <certificate> | openssl base64 -d \  
| dd bs=1 skip=24 2>/dev/null | openssl sha1 -binary | openssl base64
```

The resulting value is the calculated `DnQualifier` of the public key in the input certificate. Confirm that when this calculation is performed on the public key in the subject certificate, the calculated value is equal to the `DnQualifier` present in the `Subject` field. Confirm that when this calculation is performed on the public key in the issuer certificate, the calculated value is equal to the `DnQualifier` present in the `Issuer` field of the subject certificate. A `DnQualifier` that does not match the calculated value of the corresponding certificate's public key shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Reference Document ID	Reference Document Section(s)
[SMPTE-430-2]	

Test Equipment
openssl

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Released May 8 2009



## 2.1.12. Organization Name Field

### Objective

Verify that exactly one instance of the `OrganizationName` field is present in the `Issuer` and `Subject` fields.  
 Verify that the two `OrganizationName` values are identical.

### Procedures

The presence of the `OrganizationName` in the `Subject` and `Issuer` fields can be verified by using the `openssl` command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The `OrganizationName` values are in the `Subject` and `Issuer` fields in the certificate as indicated by **5** and **10** in the example certificate. Confirm that the Organization name, the value specified as "O=<organization-name>", is the same in both fields. Non-identical Organizational name values in the `Subject` and `Issuer` fields shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 2.1.13. OrganizationUnitName Field

### Objective

Verify that exactly one instance of the OrganizationUnitName (OU) value is present in the Issuer and Subject fields.

### Procedures

The presence of the OrganizationUnitName in the Subject and Issuer fields can be verified by using the **openssl** command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The OrganizationUnitName values are in the Subject and Issuer fields in the certificate as indicated by **5** and **10** in the example certificate. The absence of an OrganizationUnitName in either the Subject or Issuer fields of the certificate shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.14. Entity Name and Roles Field

### Objective

Verify that the `CommonName` (CN) is present exactly once in both the `Subject` and `Issuer` fields. Also verify that the `CommonName` fields contain a physical identification of the entity (i.e., make, model, or serial number, for devices). For leaf certificates (i.e., certificate authority is set to `False`), verify that at least one role is specified and that it is the role expected for the certificate.

### Procedures

The presence of the `CommonName` in the `Subject` and `Issuer` fields can be verified by using the `openssl` command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

The `CommonName` values are in the `Subject` and `Issuer` fields in the certificate as indicated by **5** and **10** in the example certificate. Confirm that the `CommonName`, the value specified as "CN=<common-name>" is present only once and that it contains information that identifies the entity. For leaf certificates, confirm that the common name specifies at least one role and that it is correct for the certificate. The absence of the `CommonName` value in either the `Subject` or `Issuer` fields shall be cause to fail this test. For leaf certificates, the absence of a role designation shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.15. Unrecognized Extensions

### Objective

Verify that any X.509v3 extensions in the certificate that are not specified in [SMPTE 430-2] (unrecognized extensions) are not marked critical.

### Procedures

The list of X.509v3 extensions in a certificate can be viewed by using the **openssl** command to display the certificate information as described in Example 2.1, e.g.:

```
$ openssl x509 -text -noout -in <certificate>
```

Of the X.509v3 extensions listed in the certificate, only "Basic Constraints" (indicated by **19**) must be marked critical, and then only for signer certificates (certificates that have CA:TRUE). "Basic Constraints" may be marked critical for leaf certificates. "Key Usage" (indicated by **17**) may be marked critical. No other X.509v3 extensions may be marked critical. A signer certificate with a "Basic Constraints" section that is not marked critical shall be cause to fail this test. A Certificate that has any X.509v3 extension marked critical other than Basic Constraints and Key Usage shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.16. Signature Validation

### Objective

Using the issuer's public key, verify that the signature contained in the certificate is valid.

### Procedures

For this operation to be successful, validation must be performed down the certificate chain, from the self-signed root certificate (the CA) to the leaf certificate being validated. Certificate chain validation is recursive, so as each certificate in the chain is validated it is included as part of the validation of the next certificate. With OpenSSL, this results in a file that contains the root certificate and, incrementally, each of the signer certificates of certificate chain of the leaf certificate. This file is then used to validate the signature on the leaf certificate. A certificate chain containing three certificates can be validated by following these steps:

1. Verify that the CA certificate signature is valid
2. Verify that the CA's signature on the signer's certificate is valid.
3. Verify that the signer's signature on the leaf certificate is valid.

This example uses **openssl** to validate each certificate, and the unix command '**cat**' to append each successive certificate to a single file. This file is specified to **openssl** using the **-CAfile** option.

```
$ openssl verify -CAfile caroot.pem caroot.pem
caroot.pem: OK
$ cp caroot.pem certchain.pem
$ openssl verify -CAfile certchain.pem signer.pem
signer.pem: OK
$ cat signer.pem >> certchain.pem
$ openssl verify -CAfile certchain.pem leaf.pem
leaf.pem: OK
```

Error messages from OpenSSL indicate that a certificate in the chain did not validate, and that the chain is not valid. Error messages that indicate that the certificate chain is not valid shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Equipment
openssl

## 2.1.17. Certificate Chains

### Objective

Verify that the following conditions are true for a given certificate chain:

1. The certificate chain is complete, i.e., for each certificate specified in an `Issuer` field, there is a corresponding certificate whose `Subject` field matches that `Issuer` field.
2. For each certificate in the chain, the validity periods of any child certificates are completely contained within the validity period of the parent certificates.
3. That the root certificate (i.e., a self-signed certificate where the `CA-flag` is true) is a valid root certificate.

### Procedures

A complete certificate chain starts with a leaf certificate and ends with a self-signed (CA root) certificate. Between the leaf certificate and the CA root certificate there should be one or more signer certificates. A leaf certificate is signed by a signer certificate, and the signer certificate is identified by its `DnQualifier` in the "Issuer" field of the leaf certificate. In a chain of three certificates, the signer certificate is in turn signed by the CA root certificate, which is similarly identified by its `DnQualifier` in the `Issuer` field of the signer's certificate. The CA root certificate is self-signed and has its own `DnQualifier` in both the `Subject` and `Issuer` fields.

To verify that the certificate chain is complete, confirm that the certificates corresponding to the `Issuer DnQualifiers` of each of the certificates is present, as explained in Section 2.1.11: `Public Key Thumbprint`. A certificate chain that does not contain all of the certificates matching the `DnQualifiers` specified in the `Issuer` fields of the certificates means the chain is not complete and shall be cause to fail this test.

The validity period of a certificate can be viewed using the procedure described in Section 2.1.7: `Validity Field`. Confirm that for each certificate in the chain, the signer certificate's validity period completely contains the validity period of the signed certificate. A certificate that has a validity period that extends beyond the validity period of its signer (either starting before, or ending after, the validity period of its signer) shall be cause to fail this test.

To confirm that the CA root certificate is a valid root certificate:

1. Verify that the `DnQualifier` in the `Issuer` field is the same as the `DnQualifier` in the `Subject` field as described in Section 2.1.11: `Public Key Thumbprint`.
2. Confirm that the `Certificate Authority` value in the `Basic Constraints` field is true and the `path length` value is a number greater than zero, as described in Section 2.1.10: `Basic Constraint Field`.
3. Confirm that the `X.509v3 Key Usage` contains "Certificate Sign" as described in Section 2.1.9: `KeyUsage Field`.

A CA certificate that is not self signed, or does not have a non-negative `path length` of zero or greater, or that does not have the `basic constraints extension` marked critical and containing `CA:TRUE`, shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

<b>Test Equipment</b>
openssl

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Discontinued May 8 2009

## 2.2. Certificate Decoder Behavior

### 2.2.1. ASN.1 DER Encoding Check

#### Objective

Verify that a certificate is rejected by the decoding device if it contains syntax errors or does not conform to the ASN.1 DER (Distinguished Encoding Rules) format.

#### Procedures

For the malformed certificate below, perform an operation with the device under test using a malformed certificate. Verify that the operation fails. A successful operation using a malformed certificate is cause to fail this test.

1. A certificate encoded as BER (*KDS-chain-c3-BER-enc*, *IMB-chain-a3-BER-enc*)

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-2]	

Test Material
KDS-chain-c3-BER-enc
chain-c1-root
chain-c3-root
IMB-chain-a3-BER-enc
chain-a3-root
chain-b1-root



## 2.2.2. Missing Required Fields

### Objective

Verify that certificates with missing required fields are rejected by a device under test.

### Procedures

For each of the malformations below, perform an operation on the device with the certificate that contains that malformation. Verify that the operation fails. A successful operation using a malformed certificate is cause to fail this test.

- missing SignatureAlgorithm field (i.e, *chain-a3-no-saf*, *chain-a3-no-saf*) - reject
- missing SignatureValue field (*chain-a3-no-svf*, *chain-a3-no-svf*) - reject
- missing Version field (*chain-a3-no-ver*, *chain-a3-no-ver*) - reject
- missing SerialNumber field (*chain-a3-no-sn*, *chain-a3-no-sn*) - reject
- missing Signature field (*chain-a3-no-sig*, *chain-a3-no-sig*) - reject
- missing Issuer field (*chain-a3-no-issuer*, *chain-a3-no-issuer*) - reject
- missing Subject field (*chain-a3-no-subject*, *chain-a3-no-subject*) - reject
- missing SubjectPublicKeyInfo field (*chain-a3-no-spki*, *chain-a3-no-spki*) - reject
- missing Validity field (*chain-a3-no-val-f*, *chain-a3-no-val-f*) - reject
- missing AuthorityKeyIdentifier field (*chain-a3-no-aki-f*, *chain-a3-no-aki-f*) - reject
- missing KeyUsage field (*chain-a3-no-keyuse*, *chain-a3-no-keyuse*) - reject
- missing BasicConstraint field (*chain-a3-no-basic*, *chain-a3-no-basic*) - reject

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
chain-a3-no-saf
chain-a3-no-svf
chain-a3-no-ver
chain-a3-no-sn
chain-a3-no-sig
chain-a3-no-issuer
chain-a3-no-subject

Test Material
chain-a3-no-spki
chain-a3-no-val-f
chain-a3-no-aki-f
chain-a3-no-keyuse
chain-a3-no-basic
chain-c1-root
chain-c3-root
chain-a3-no-aki-f
chain-a3-no-basic
chain-a3-no-issuer
chain-a3-no-keyuse
chain-a3-no-saf
chain-a3-no-sig
chain-a3-no-sn
chain-a3-no-spki
chain-a3-no-subject
chain-a3-no-svf
chain-a3-no-val-f
chain-a3-no-ver

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 2.2.3. PathLen Check

#### Objective

Verify that, if the Certificate Authority attribute of the BasicConstraint field is True, the PathLenConstraint value is present and is either zero or positive. Verify that if the certificate authority attribute of the BasicConstraint field is False, the PathLenConstraint field is absent or set to zero.

#### Procedures

1. Perform an operation on the device under test using a leaf certificate with a PathLen greater than zero (0). Verify that the operation fails. A successful operation using a certificate with an incorrect Path Length is cause to fail this test.
2. Perform an operation on the device under test using a leaf certificate with a PathLen that is negative. Verify that the operation fails. A successful operation using a certificate with an incorrect Path Length is cause to fail this test.
3. Perform an operation on the device under test using a signer certificate that does not contain a PathLen (PathLen absent). Verify that the operation fails. A successful operation using a certificate with an incorrect Path Length is cause to fail this test.
4. Perform an operation on the device under test using a signer certificate that contains a PathLen that is negative. Verify that the operation fails. A successful operation using a certificate with an incorrect Path Length is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
chain-a3-path-1
chain-a3-path-2
chain-a3-path-3
chain-a3-path-4
chain-a3-path-5
chain-a3-path-6
chain-a3-path-7
chain-c3-root
chain-a3-path-1
chain-a3-path-2
chain-a3-path-3
chain-a3-path-4
chain-a3-path-5
chain-a3-path-6

Test Material
chain-a3-path-7
chain-a3-root

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Used May 8 2009

## 2.2.4. OrganizationName Match Check

### Objective

Verify that the certificate is rejected by the device if the OrganizationName in the subject and issuer fields do not match.

### Procedures

Perform an operation on the device with a certificate that has mismatched OrganizationName values in the Subject and Issuer fields. Verify that the operation fails. A successful operation using a malformed certificate is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
chain-a3-org-name
chain-c3-root
chain-a3-org-name
chain-a3-root

## 2.2.5. Certificate Role Check

### Objective

Verify that when the validation context includes a desired role, a device under test rejects a leaf certificate with a role that is different than the role expected.

### Procedures

Perform an operation on the device under test using a certificate with a role that is not permitted to perform the operation. Verify that the operation fails. A successful operation using a certificate with an incorrect role in a validation context that requires the correct role is cause to fail this test.

- Certificate Authority is False and no role specified in CommonName (*chain-a3-role-1*, IMB-chain-a3-role-1) - reject
- Distribution Root Certificate without a distributor role, Remote SPB root Certificate with a role other than SMS role (*KDS chain-a3-role-2*, *chain-a3-role-2*) - reject

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
chain-a3-role-1
KDS chain-a3-role-2
chain-c3-root
chain-a3-role-1
chain-a3-role-2
chain-a3-root

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 2.2.6. Validity Date Check

### Objective

Verify that the certificate is rejected if it is not valid at the desired time (according to the validation context, e.g., time of playback).

### Procedures

Perform an operation on the device with a certificate that is not valid. Verify that the operation fails. A successful operation using a certificate at a time outside of its validity period is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
chain-a3-date-exp
chain-c3-root
chain-a3-date-exp
chain-a3-root

## 2.2.7. Signature Algorithm Check

### Objective

Verify that a certificate is rejected by a device under test if the signature algorithms in the certificate body and the signature are not `sha256WithRSAEncryption`.

### Procedures

Perform an operation on the device with a certificate that has mismatched or incorrect signatures for each of the following types of signature errors. Verify that the operation fails. A successful operation using an incorrectly signed certificate is cause to fail this test.

- Signature algorithm of the signature not `sha256WithRSAEncryption` (*chain-a3-osig-type, chain-a3-iosig-type*) - reject
- Signature algorithm of the certificate not `sha256WithRSAEncryption` (*chain-a3-isig-type, chain-a3-isig-type*) - reject
- Signature algorithms identical, but not `sha256WithRSAEncryption` (*chain-a3-iosig-type, chain-a3-osig-type*) - reject

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
chain-a3-osig-type
chain-a3-isig-type
chain-a3-iosig-type
chain-c3-root
chain-a3-iosig-type
chain-a3-isig-type
chain-a3-osig-type
chain-a3-root



## 2.2.8. Public Key Type Check

### Objective

Verify that the certificate is rejected if the subject's Public Key is not a 2048 bit RSA key with an exponent of 65537.

### Procedures

For each of the types of incorrect public keys below, perform an operation on the device with the certificate that has an public key that is not correct. Verify that the operation fails. A successful operation using a certificate with an incorrect public key is cause to fail this test.

- Public Key not an RSA Key (*chain-a3-no-rsa*, *chain-a3-no-rsa*) - reject
- RSA Public Key Length only 1024 bit (*chain-a3-short-rsa*, *chain-a3-short-rsa*) - reject
- Public Key Exponent other than 65537 (*chain-a3-bad-exp*, *chain-a3-bad-exp*) - reject

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
chain-a3-no-rsa
chain-a3-short-rsa
chain-a3-bad-exp
chain-c3-root
chain-a3-no-rsa
chain-a3-bad-exp
chain-a3-short-rsa
chain-a3-root

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 2.2.9. Issuer Certificate Presence Check

### Objective

Verify that the certificate is rejected if the issuer's certificate cannot be located by looking it up using the value of the AuthorityKeyIdentifier X.509v3 extension.

### Procedures

Perform an operation on the device under test using certificates that do not include the certificate's signer specified by the AuthorityKeyIdentifier. Verify that the operation fails. A successful operation using a certificate without the certificate signer present is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
KDM without AuthorityKey certificate

---

# Chapter 3. Key Delivery Messages

This chapter contains tests for Key Delivery Messages (KDM). The test procedures in this chapter are organized into three groups: tests that evaluate a KDM's compliance to [SMPTE 430-3], tests that evaluate a KDM's compliance to [SMPTE 430-1], and tests that evaluate the behavior of devices that decode KDMs. The KDM Decoder tests are in this section because they are not specific to any particular type of system. All d-cinema devices that decode KDMs must behave in the manner described by these tests.

Before diving in to testing KDM files, we will first introduce XML and provide some examples of KDM documents.

## 3.1. eXtensible Markup Language

XML is a file metaformat: a file format for creating file formats. Many of the files that comprise a d-cinema composition (e.g., a feature or trailer), are expressed in XML. While the various d-cinema file formats represent different concepts within the d-cinema system, the arrangement of data within the files is syntactically similar for those files that use XML. This section will provide an overview of XML as used for d-cinema applications. Readers looking for more detailed technical information are referred to the home of XML at <http://www.w3.org>.

### 3.1.1. XML Documents

The main unit of data storage in an XML document is the XML *element*. XML elements are expressed in a document using *tags*; strings of human-readable text enclosed between less-than (<) and greater-than (>) characters. An XML *document* is an element that is meant to be interpreted as a complete unit. Every XML document consists of a single XML element having zero or more (usually hundreds more) elements inside. XML documents may be stored as file, transmitted over networks, etc. The following example shows a very simple XML element, rendered as a single tag:

```
<Comment />
```

By itself, this XML element is a complete, though very uninteresting, XML document.

To be more useful, our example element needs some data, or *content*. XML content may include unstructured text or additional XML elements. Here we have expanded the element to contain some text:

```
<Comment>The quick brown fox...</Comment>
```

Notice that when an XML element has content, the content is surrounded by two tags, in this case <Comment> and </Comment>. The former is an *opening* tag, the latter a *closing* tag.

We now have some data inside our element. We could help the reader of our example XML document by indicating the language that the text represents (these same characters could of course form words from other languages). The language of the text is *metadata*: in this case, data about the text. In XML, metadata is stored as sets of key/value pairs, or *attributes*, inside the opening tags. We will add an attribute to our example element to show some metadata, in this case we are telling the reader that the text is in English:

```
<Comment language="en">The quick brown fox...</Comment>
```

The following example shows an actual d-cinema data structure (there is no need to understand the contents of this example as this particular structure is covered in more detail in Section 4.2.1.):

#### Example 3.1. Packing List Example (Partial)

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

```

<PackingList xmlns="http://www.smpte-ra.org/schemas/429-8/2007/PKL">
  <Id>urn:uuid:59430cd7-882d-48e8-a026-aef4b6253dfc</Id>
  <AnnotationText>Perfect Movie DCP</AnnotationText>
  <IssueDate>2007-07-25T18:21:31-00:00</IssueDate>
  <Issuer>user@host</Issuer>
  <Creator>Packaging Tools v1.0</Creator>
  <AssetList>
    <Asset>
      <Id>urn:uuid:24d73510-3481-4ae5-b8a5-30d9eeced9c1</Id>
      <Hash>AXufMKY7NyZcfSXQ9sCZ1s5dSyE=</Hash>
      <Size>32239753</Size>
      <Type>application/mxf</Type>
      <AnnotationText>includes M&E</AnnotationText>
    </Asset>
  </AssetList>
</PackingList>

```

### 3.1.2. XML Schema

You may have noticed that the basic structure of XML allows the expression of almost unlimited types and formats of information. Before a device (or a person) can read an XML document and decide whether it is semantically correct, it must be possible for the reader to know what the document is expected to contain.

The XML standard dictates some initial requirements for XML documents. The document shown in Example 3.1 above illustrates some of these requirements:

1. Element tags must be correctly nested: an element must be closed in the same scope in which it was opened. For example, the following XML fragment shows incorrect nesting of the `Element3` element (it should close before `Element2` closes, not after).

```

<Element1>
  <Element2>
    <Element3>
  </Element2>
</Element3>
</Element1>

```

2. The document may not contain special characters in unexpected places. For example, the `&`, `<` and `>` characters may not appear except in certain cases. Special encodings must be used to use these characters literally within an XML document.

A document which meets these requirements is said to be *well formed*. All XML documents must be well formed. An XML *parser* (a program that reads XML syntax) will complain if you give it XML that is not well-formed. Well-formedness, however, does not help us understand *semantically* what's in an XML document. To know the meaning of a particular XML structure, we have to have a description of that structure.

The structure and permitted values in an XML document can be defined using XML Schema. There are other languages for expressing the content model of an XML document, but XML Schema is the standard used by the SMPTE specifications for d-cinema. XML Schema is a language, expressed in XML, which allows the user to define the names of the elements and attributes that can appear in a XML document. An XML Schema can also describe the acceptable contents of and combinations of the XML elements.

Given an XML Schema and an XML document, a *validating* XML parser will report not only errors in syntax but also errors in the use and contents of the elements defined by the schema. Throughout this document, we will use the **schema-check** program to test XML documents. The command takes the instance document and one or more schema documents as arguments:

```
$ schema-check <input-file> smpte-430-3-2007.xsd
```

If this command returns without errors, the XML document can be said to be both well-formed and *valid*.

Some XML documents are defined using more than one schema. In these cases, you can supply the names of any number of schemas on the command line:

```
$ schema-check <input-file> smpte-430-3-2007.xsd smpte-430-1-2007.xsd
```

### 3.1.3. XML Signature Validation

*XML Signature* is a standard for creating and verifying digital signatures on XML documents. Digital signatures are used to allow recipients of Composition Playlists, Packing Lists and Key Delivery Messages (KDM) to *authenticate* the documents; to prove that the documents were signed by the party identified in the document as the document's signer, and that the documents have not been modified or damaged since being signed.

The **checksig** program (distributed with the XML Security library) can be used to test the signature on an XML document. The program is executed with the name of a file containing a signed XML document:

#### Example 3.2. checksig execution

```
$ checksig test-kdm.xml  
Signature verified OK!
```

The program expects that the first certificate in the `<KeyInfo>` element is the signer. This has two implications:

- The program will fail if the signer is not the first (SMPTE standards allow any order)
- The program does not check the entire certificate chain

To solve the first problem, the **dsig-cert.py** (see Section C.8) can be used to re-write the XML document with the signer's certificate first in the `<KeyInfo>` element. This is demonstrated in the following example:

#### Example 3.3. dsig-cert.py execution

```
$ dsig-cert.py test-kdm.xml >tmp.xml  
$ checksig tmp.xml  
Signature verified OK!
```

The second problem is solved by extracting the certificates from the document's XML Signature data and validating them directly with **openssl**. This procedure is the subject of the next section.

#### 3.1.3.1. Extracting Certificates from an XML Document

In order to test certificates separately from the XML document in which they are embedded, it is necessary to manually extract them into separate PEM files (see [RFC-1421]). A PEM file contains a certificate (more than one if desired, but we're not going to do that just yet) as a DER-encoded binary string which is then encoded using Printable Encoding (see [RFC-1421]). The encoded text is prefixed by the string `-----BEGIN CERTIFICATE-----` followed by a newline. The encoded text is followed by the string `-----END CERTIFICATE-----`. An example of this format can be seen below. Note that the Printable Encoding has newlines after every 64 characters.

### Example 3.4. An X.509 certificate in PEM format

```
-----BEGIN CERTIFICATE-----
MIIEdzCCA1+gAwIBAgICNBowDQYJKoZIhvcNAQELBQAwwGQYXQxGTAXBgNVBAoTEC5j
YS5jaW5lY2VydC5jb20xLDAqBgNVBAsTIy5yYS0xYi5yYS0xYS5zNDMwLTUyY2Eu
Y2luZW50cnQuY29tMRlweAYDVQDEwkuY2MtYWRtaW4xJTAjBgNVBC4THGNwSmxw
NDBCM0hqSG9kOG9JWnpsVi9DU0xmND0wIBcNMDcwMTE1MjI0OTQ0WhgPMjAwODAx
MTUyYjQ5NDJhMjI0MjI0MjI0MjI0MjI0MjI0MjI0MjI0MjI0MjI0MjI0MjI0MjI0
EywuY2MtYWRtaW4ucmEtMWEucmEtMWEucmEtMWEucmEtMWEucmEtMWEucmEtMWEucm
MA4GA1UEAxMHU00ud3MtMTE1MCMGA1UELhMEdC8zQ2xNWjdjQWRGUUnhnamlTRTFn
NGY4NUhNPTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAA0BejWa3Lg+Y
uvTYhCaFy0ET6zH6XrB3rLr1beMrrTuUMCX0YSma7m3Z01Bd/HQrJxyq6hJmPGu
auxwWiF4w+AAjBRp4eSiAt8srACcEmUyqGHwPLoaKVEaHXSOY8gJp1kZwqGwoR40
RQusfAb2/L76+R1MUyACoJuR6k4k0BW3bjEE4E76KKR4k5K580d7uFf5G86GhGFU
AfXHJXboqzHnxQHAmldKNaSskxWrW8GrX43+2ZZUHM2ZKe0Ps/9g2gCRZ6eYaIm4
UF+szH0EUY0Mbx4poqn+SZFrUWtEoWcDM6PSTTgCQVOQ1BtzD11BQoNQGOJcd73N
9f5MfGioWmKcAwEAAa0B5zCB5DALBgNVHQ8EBAMCBLAwDAYDVR0TAAQH/BAIwADAd
BgNVHQ4EFgQut/3ClMz7bAdFRxgjmSElg4f85HMwgacGA1UdIwSbnzCBnIAUcPJl
p40B3HjHod8oIzZlV/CSLf6hf6R9MHsxGTAXBgNVBAoTEC5jYS5jaW5lY2VydC5j
b20xJjAkBgNVBAsTSH5yYS0xYi5yYS0xYS5zNDMwLTUyY2EuY2luZW50cnQuY29tMQ8wDQYD
VQDEwYucmEtMWEucmEtMWEucmEtMWEucmEtMWEucmEtMWEucmEtMWEucmEtMWEucm
Zz2CAwDpztANBgkqhkiG9w0BAQsFAAOCAQEaowjAFQsyoKto7+WBeF9HuCRpKkxk
6qMgXzGafJFRk/pi7CjnfjxvWukJq4HWgWHPXsGff/RTp08naV1UHNe71sDYV2Fb
MOSFRi2OrRwZexO9SBKQHLZ7ZdLU+6GIHXKjmp9DiofUNOqvZPQnvwG/Cm084CpG
Kl4ktxtOghczEiJCK2KISsgOU6NK4cmcFfMjuklTwmD5C6TvaawkvcNJQcldjUw
TWbvd+Edf9wkHNVBERR9lbcGwr16C5BVQZtFBJAU++3guL/4Qn41keU/gmR6o99S
UQ+T344CBSIy06ztiWziuxoNoXfy12DTsepB+QShmuhsScrfv0Q9bB5hw==
-----END CERTIFICATE-----
```

Within an XML document signed using XML Signature, certificates are stored in `<dsig:X509Certificate>` elements. These elements can be found at the end of the document, within the `</dsig:Signature>` element. The encoding method for storing certificate data in XML Signature is virtually identical to PEM. The Base64 encoding (see [RFC-2045]) uses the same mapping of binary data to text characters, but the line length is not limited as with PEM.

It is a relatively easy task to use a text editor to copy and paste certificate data from an XML document:

1. Open a new text editor window, and paste `-----BEGIN CERTIFICATE-----`, then press the Enter key. Note that the number of '-' (dash) characters on either side of the `BEGIN CERTIFICATE` label is five (5).
2. Copy the content of the selected `<dsig:X509Certificate>` element (but not the element tags) from the KDM and paste it into the new editor window. The cursor should now be positioned at the last character of the certificate; press the Enter key.
3. Paste `-----END CERTIFICATE-----` at the end of the new editor window and press the Enter key.
4. Note again that Printable Encoding lines in PEM format files must be no more than 64 characters in length. If the Base64 certificate string copied from the KDM contains long lines, manually break the lines using the cursor and the Enter key.
5. Save the editor's contents to a file, usually with a `.pem` suffix.

In most cases the procedure given above can be automated using the `dsig-extract.py` program (see Section C.9). As shown below, the `-p` option can be used to provide a prefix for the automatically-generated filenames. In this example, the input document contained four certificates.

### Example 3.5. dsig-extract.py execution

```
$ dsig-extract.py -p my_prefix_ test-kdm.xml
```

```
$ ls my_prefix_*
my_prefix_1.pem
my_prefix_2.pem
my_prefix_3.pem
my_prefix_4.pem
```

You can test that the certificate has been correctly extracted by using **openssl** to view the contents of the certificate file:

```
$ openssl x509 -text -noout -in <certificate-file.pem>
```

The output from this command should look similar to Example 2.1: D-Cinema Certificate.

To validate a complete chain of extracted certificates, use the procedure in Section 2.1.16.

## 3.2. Key Delivery Message Example

The Key Delivery Message (KDM) is an XML document that contains cryptographic information necessary to reproduce an encrypted composition. A KDM also contains metadata about the cryptographic information, such as the validity period and the associated Composition Playlist (CPL). The format of the KDM file is specified by [SMPTE-430-1]. A KDM is a type of Extra-Theater Message (ETM), as specified by [SMPTE-430-3].

The following examples show the elements of the KDM that will be examined during the procedures. Each example is followed by a list of descriptive text that describes the various features of the KDM called out in the examples. These features will be referred to from the test procedures.

### Example 3.6. KDM - AuthenticatedPublic area

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?> 1
<DCinemaSecurityMessage xmlns="http://www.smpte-ra.org/schemas/430-3/2006/ETM" 2
  xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:enc="http://www.w3.org/2001/04/xmlenc#">
  <AuthenticatedPublic Id="ID_AuthenticatedPublic"> 3
    <MessageId>urn:uuid:b80e668c-a175-4bc7-ae48-d3a19c8fce95</MessageId> 4
    <MessageType>http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type</MessageType> 5
    <AnnotationText>Perfect Movie KDM</AnnotationText> 6
    <IssueDate>2007-07-24T17:42:58-00:00</IssueDate> 7
    <Signer> 8
      <dsig:X509IssuerName>dnQualifier=wBz3yptkPxbHI/\+LUUeH5R6rQfI=,CN=.cc-admin-x,
        OU=.cc-ra-1a.s430-2.ca.example.com,O=.ca.example.com</dsig:X509IssuerName>
      <dsig:X509SerialNumber>6992</dsig:X509SerialNumber>
    </Signer>
    <RequiredExtensions>
      <KDMRequiredExtensions xmlns="http://www.smpte-ra.org/schemas/430-1/2006/KDM">
        <Recipient> 9
          <X509IssuerSerial>
            <dsig:X509IssuerName>dnQualifier=wBz3yptkPxbHI/\+LUUeH5R6rQfI=,CN=.cc-admin-x,
              OU=.cc-ra-1a.s430-2.ca.serverco.com,O=.ca.serverco.com</dsig:X509IssuerName>
            <dsig:X509SerialNumber>8992</dsig:X509SerialNumber> 10
          </X509IssuerSerial>
          <X509SubjectName>dnQualifier=83R40icxCejFRR6Ij6iwdf2faTY=,CN=SM.x_Mastering,
            OU=.cc-ra-1a.s430-2.ca.example.com,O=.ca.example.com</X509SubjectName> 11
        </Recipient>
        <CompositionPlaylistId> 12
          urn:uuid:20670ba3-d4c7-4539-ac3e-71e874d4d7d1
        </CompositionPlaylistId>
        <ContentTitleText>Perfect Movie</ContentTitleText> 13
        <ContentKeysNotValidBefore>2007-07-24T17:42:54-00:00</ContentKeysNotValidBefore> 14
        <ContentKeysNotValidAfter>2007-08-23T17:42:54-00:00</ContentKeysNotValidAfter> 15
```

```

<AuthorizedDeviceInfo>
  <DeviceListIdentifier>urn:uuid:d47713b9-cde1-40a9-98fe-22ef172723d0</DeviceListIdentifier>
  <DeviceList> 16
    <CertificateThumbprint>jk4Z8haFhqCGAVbClW65jVSOib4=</CertificateThumbprint> 17
  </DeviceList>
</AuthorizedDeviceInfo>
<KeyIdList> 18
  <TypedKeyId>
    <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDIK</KeyType> 19
    <KeyId>urn:uuid:15e929b3-1d86-40eb-875e-d21c916fdd3e</KeyId> 20
  </TypedKeyId>
  <TypedKeyId>
    <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDAK</KeyType>
    <KeyId>urn:uuid:ca8f7756-8c92-4e84-a8e6-8fab898934f8</KeyId>
  </TypedKeyId>
[remaining key IDs omitted for brevity]
</KeyIdList>
<ForensicMarkFlagList>
  <ForensicMarkFlag> 21
    http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-audio-disable
  </ForensicMarkFlag>
</ForensicMarkFlagList>
</KDMRequiredExtensions>
</RequiredExtensions>
<NonCriticalExtensions/>
</AuthenticatedPublic>

```

### KDM AuthenticatedPublic area descriptions

- 1** XML Declaration. This specifies the version of the XML standard to which the document conforms, and the character encoding of the document.
- 2** The root DCinemaSecurityMessage element. This element contains the XML namespace declaration for a KDM as specified in [SMPTE-430-1].
- 3** The beginning of the AuthenticatedPublic section of the KDM.
- 4** The Unique Universal ID (UUID) of the KDM. This is used to uniquely identify the asset map.
- 5** The type of message, in this case a KDM.
- 6** An annotation text describing the contents or purpose of the KDM.
- 7** The date the KDM was issued.
- 8** The portion of the KDM that holds information about the certificate used to sign the KDM.
- 9** The portion of the KDM that contains information about the recipient (target) certificate.
- 10** The serial number of the recipient certificate.
- 11** The Subject Name information from the recipient certificate.
- 12** The UUID of the CPL used to create the KDM.
- 13** The ContentTitleText from the CPL used to create the KDM.
- 14** The starting validity date of the KDM.
- 15** The ending validity date of the KDM.
- 16** Device list. This list contains the list of certificates thumbprints authorized for use with at least a portion of the KDM.
- 17** A certificate thumbprint in the device list.
- 18** The list of KeyIDs and their associated type.
- 19** The type of key represented by the KeyID.
- 20** The KeyID.
- 21** This flag determines whether forensic marking is enabled or disabled. The ForensicMarkFlagList may contain multiple instances of ForensicMarkFlag.

### Example 3.7. KDM - AuthenticatedPrivate area

```

<AuthenticatedPrivate Id="ID_AuthenticatedPrivate"> 1
  <enc:EncryptedKey xmlns:enc="http://www.w3.org/2001/04/xmlenc#"> 2
  <enc:EncryptionMethod

```



```

    Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgflp"> ❸
<ds:DigestMethod xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
</enc:EncryptionMethod>
<enc:CipherData>
<enc:CipherValue> ❹
[ 256 Byte long encrypted cipherdata block omitted]
</enc:CipherValue>
</enc:CipherData>
</enc:EncryptedKey><enc:EncryptedKey xmlns:enc="http://www.w3.org/2001/04/xmlenc#">
<enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgflp">
<ds:DigestMethod xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
</enc:EncryptionMethod>
<enc:CipherData>
<enc:CipherValue>
[ 256 Byte long encrypted cipherdata block omitted]
</enc:CipherValue>
</enc:CipherData>
</enc:EncryptedKey>
<enc:EncryptedKey xmlns:enc="http://www.w3.org/2001/04/xmlenc#">
<enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgflp">
<ds:DigestMethod xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
</enc:EncryptionMethod>
<enc:CipherData>
<enc:CipherValue>
[ 256 Byte long encrypted cipherdata block omitted]
</enc:CipherValue>
</enc:CipherData>
</enc:EncryptedKey><enc:EncryptedKey xmlns:enc="http://www.w3.org/2001/04/xmlenc#">
<enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgflp">
<ds:DigestMethod xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
</enc:EncryptionMethod>
<enc:CipherData>
<enc:CipherValue>
[ 256 Byte long encrypted cipherdata block omitted]
</enc:CipherValue>
</enc:CipherData>
</enc:EncryptedKey>
[additional EncryptionKey entries omitted]
</AuthenticatedPrivate>

```

### KDM AuthenticatedPrivate area descriptions

- ❶ The start of the AuthenticatedPrivate section of the KDM.
- ❷ The EncryptedKey element indicates there is data encrypted with an RSA public key algorithm.
- ❸ The algorithm used to encrypt the data in the CipherData element.
- ❹ A 256 Byte long block of RSA encrypted data.

### Example 3.8. KDM - Signature area

```

<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"> ❶
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/> ❷
    <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/> ❸
    <dsig:Reference URI="#ID_AuthenticatedPublic"> ❹
      <dsig:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/> ❺
      <dsig:DigestValue>cnn8M41NR4jQF+9GOZiNJTlfl+C/l8lBF1juCuq9lQE=</dsig:DigestValue> ❻
    </dsig:Reference>
    <dsig:Reference URI="#ID_AuthenticatedPrivate"> ❼

```

```

    <dsig:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
    <dsig:DigestValue>TEW7tPwML2iOkIpK2/4rZbJbKggnXjAtJwe9OJSe8u4=</dsig:DigestValue>
  </dsig:Reference>
</dsig:SignedInfo>
  <dsig:SignatureValue>uH41s9odRPXzFz+BF3dJ/myG09cLSE9cLzf2C7f2Fm49P9C53T5RSeEIyqt6p511 8
z1H2q3ZJRZcZuV5VA7UkIb4z6U4CGUTU51D81L/anY1glLFddjUiDU/0nmC4uAsH
rzWQgzOTZmZd2eLo0N70DBtNhTcJZftKUN2O2ybHZaJ7Q/aBxAiCK3h/fRW/b7zM
bcbsD9/VfJFI7VQCOLYwTxq643Exj7sYGKISrjuN+MLAubG50hu74YLOtA/dmGB1
G4VeXkBBR/BEjOEoxyfFpXbZwkdoI18/Qd1JF32xpE1PlTLrJoRyjrX/6qkm9OJ
X9GyFNd8jVxdYNI4s1JCnQ==</dsig:SignatureValue>
  <dsig:KeyInfo>9
<dsig:X509Data>
<dsig:X509IssuerSerial>
<dsig:X509IssuerName>dnQualifier=wBz3yptkPxbHI/\+LUUEH5R6rQfI=,
  CN=.cc-admin-x,OU=.cc-ra-la.s430-2.ca.example.com,O=.ca.example.com</dsig:X509IssuerName>
<dsig:X509SerialNumber>6992</dsig:X509SerialNumber>
</dsig:X509IssuerSerial>
<dsig:X509Certificate> 10
  [PEM encoded certificate omitted]
</dsig:X509Certificate>
</dsig:X509Data>
<dsig:X509Data>
<dsig:X509IssuerSerial>
<dsig:X509IssuerName>dnQualifier=808W8oYHlf97Y8n0kdAgMU7/jUU=,
  CN=.s430-2,OU=.ca.example.com,O=.ca.example.com</dsig:X509IssuerName>
<dsig:X509SerialNumber>50966</dsig:X509SerialNumber>
</dsig:X509IssuerSerial>
<dsig:X509Certificate>
  [PEM encoded certificate omitted]
</dsig:X509Certificate>
</dsig:X509Data>
<dsig:X509Data>
<dsig:X509IssuerSerial>
<dsig:X509IssuerName>dnQualifier=808W8oYHlf97Y8n0kdAgMU7/jUU=,
  CN=.s430-2,OU=.ca.example.com,O=.ca.example.com</dsig:X509IssuerName>
<dsig:X509SerialNumber>13278513546878383468</dsig:X509SerialNumber>
</dsig:X509IssuerSerial>
<dsig:X509Certificate>
  [PEM encoded certificate omitted]
</dsig:X509Certificate>
</dsig:X509Data>
</dsig:KeyInfo>
</dsig:Signature>
</DCinemaSecurityMessage>

```

### KDM Signature area descriptions

- 1 Start of the signature section of the KDM.
- 2 The canonicalization algorithm of the signature.
- 3 Specifies the signature algorithm (RSA) and the digest algorithm (SHA-256) of the signature.
- 4 The AuthenticatedPublic reference element.
- 5 The method used to create the digest of the AuthenticatedPublic portion of the KDM.
- 6 The digest of the AuthenticatedPublic portion of the KDM.
- 7 The AuthenticatedPrivate reference element.
- 8 The RSA encrypted form of the two digests.
- 9 The section of the signature portion that contains the singer certificate and its certificate chain.
- 10 The certificate used to sign the KDM.

Since the KDM carries encrypted data, a tool that can decrypt the encrypted portions of the KDM has been provided in Section C.1. **kdm-decrypt** takes two arguments, a KDM and the RSA private key that corresponds to the certificate to which the KDM was targeted, and displays the contents of the encrypted section. Here is an example of **kdm-decrypt** and the resulting output:

### Example 3.9. kdm-decrypt Usage and Output

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
CipherDataID: fldc124460169a0e85bc300642f866ab 1
SignerThumbprint: q50qr6GkfG6W2HzcBTee5m0Qjzw= 2
  CPL Id: 119d8990-2e55-4114-80a2-e53f3403118d 3
  Key Id: b6276c4b-b832-4984-aab6-250c9e4f9138 4
  Key Type: MDIK 5
  Not Before: 2007-09-20T03:24:53-00:00 6
  Not After: 2007-10-20T03:24:53-00:00 7
  Key Data: 7f2f711f1b4d44b83e1dd1bf90dc7d8c 8
```

#### kdm-decrypt output descriptions

- 1 The CipherData ID. This value is defined in [SMPTE-430-1].
- 2 Thumbprint of the certificate that signed the KDM.
- 3 The UUID of the CPL associated with this KDM.
- 4 The KeyID that corresponds to the key contained in this EncryptedKey cipherblock.
- 5 The type of key contained in this EncryptedKey cipherblock.
- 6 The beginning of validity period of the key.
- 7 The end of validity period of the key.
- 8 The encryption key.

## 3.3. ETM features

### 3.3.1. ETM Structure

#### Objective

Verify that the ETM portion of the KDM validates against the ETM schema in [SMPTE-430-3].

#### Procedures

To verify that the ETM defined elements of the KDM are well formed, validate the KDM against the ETM schema in [SMPTE-430-3], use the procedure described in Section 1.3, i.e.,

```
$ schema-check smpte-430-3-2007.xsd <input-file>
schema validation successful
```

If the KDM is not valid or well formed, the program will report an error. A reported error is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-3]	9.8

Test Equipment
schema-check Text Editor

### 3.3.2. ETM Validity Date Check

#### Objective

Verify that the signer's certificate chain was valid at the date specified in the <IssueDate> element in the <AuthenticatedPublic> area of the KDM.

#### Procedures

1. Extract each of the certificates in the signer's certificate chain from the KDM using a text editor, then, using the process described in Section 2.1.16: Signature Validation, validate the certificate chain. Validation failure of the certificate chain is cause to fail this test.
2. Once the certificate chain has been successfully validated, view the signer certificate in text form using the openssl command as described in Example 2.1: D-Cinema Certificate. Locate the Validity section of the certificate as indicated by 6 in the example certificate.
3. Using a text editor, view the contents of the KDM and locate the <IssueDate> element as shown in 7 of Example 3.6: KDM - AuthenticatedPublic area.
4. Compare the Not Before and Not After values of the signer certificate to the date in the <IssueDate> element of the KDM and confirm that it is within the date range. An <IssueDate> value outside the date ranges of the certificate is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Equipment
Text Editor openssl

### 3.3.3. ETM Signer Element

#### Objective

Verify that the certificate chain in the <Signer> element of the KDM is valid.

#### Procedures

1. Extract each of the certificates in the signer's certificate chain from the KDM using a text editor as described in Section 1.3.
2. Using the process described in Section 2.1.16: Signature Validation, validate the certificate chain. Validation failure of the certificate chain is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	
[SMPTE-430-2]	

Test Equipment
Text Editor
openssl

### 3.3.4. ETM EncryptionMethod Element

#### Objective

Verify that the Algorithm attribute of the <EncryptionMethod> for the encrypted key has the value "http://www.w3.org/2001/04/xmlenc#rsaoaep-mgf1p".

#### Procedures

Using a text editor, view the KDM and confirm that the Algorithm attribute of the <EncryptionMethod> element in the <AuthenticatedPrivate> element for each of the encrypted keys, as indicated by 3 in the example KDM, is "http://www.w3.org/2001/04/xmlenc#rsaoaep-mgf1p". Any other value in this attribute is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.3.5. ETM AnnotationText Language

#### Objective

Verify that the content of the `<AnnotationText>` element is in a human-readable language. If the optional `xml:lang` attribute is present, the language must match. If the `xml:lang` attribute is not present, the language must be English.

#### Procedures

Using a text editor, view the KDM and confirm that the `<AnnotationText>` element as indicated by 6 in the Example 3.6: KDM - AuthenticatedPublic area is a human-readable language. The presence of non-human-readable data or text in a language other than English without that language's corresponding `xml:lang` value is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor



### 3.3.6. ETM ReferenceList Element

#### Objective

Verify that the <ReferenceList> element of the <EncryptedKey> element is not present.

#### Procedures

Using a text editor, view the KDM and confirm that, for each instance of the <EncryptedKey> element, the <ReferenceList> element is not present. The presence of the <ReferenceList> element indicates that the KDM is malformed and is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

### 3.3.7. ETM SignedInfo CanonicalizationMethod Element

#### Objective

Verify that the value of the Algorithm attribute of the <CanonicalizationMethod> element of the <SignedInfo> element in the <Signature> area of the KDM is "http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments".

#### Procedures

Using a text editor, view the KDM and confirm that the value of the Algorithm attribute of the <CanonicalizationMethod> of the <SignedInfo> element of the <Signature> element is "http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments", as shown in 2 of Example 3.8: KDM - Signature area. Any other value in this attribute is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.3.8. ETM Signature Reference Elements

#### Objective

Verify that the <SignedInfo> element of the <Signature> area of the KDM contains at least two child <Reference> elements. The value of the URI attribute of each <Reference> element must correspond to the respective ID attribute of the digested element. Verify that the URI attribute of one of the <Reference> element identifies the AuthenticatedPublic portion of the KDM. Verify that the URI attribute of one of the <Reference> element identifies the AuthenticatedPrivate portion of the KDM.

#### Procedures

1. Using a text editor, view the KDM and confirm that the <SignedInfo> element of the <Signature> area of the KDM has at least two child <Reference> elements as shown in **4** and **7** of Example 3.8: KDM - Signature area. The presence of fewer than two <Reference> elements is cause to fail this test.
2. Confirm that the URI attribute of one of the <Reference> element matches the value of the ID attribute of the AuthenticatedPublic element, as shown by **4** in Example 3.8: KDM - Signature area and **3** in Example 3.6: KDM - AuthenticatedPublic area. The absence of this association in the KDM is cause to fail this test.
3. Confirm that the URI attribute of one of the <Reference> element matches the value of the ID attribute of the AuthenticatedPrivate element, as shown by **7** in Example 3.8: KDM - Signature area and **1** in Example 3.7: KDM - AuthenticatedPrivate area. The absence of this association in the KDM is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.3.9. ETM SignatureMethod Element

#### Objective

Verify that the <SignatureMethod> element of the <SignedInfo> element of the <Signature> area of the KDM contains the URI value " http://www.w3.org/2001/04/xmldsig-more#rsa-sha256".

#### Procedures

Using a text editor, view the KDM and confirm that the <SignatureMethod> element of the <SignedInfo> element of the <Signature> section of the KDM contains the URI value " http://www.w3.org/2001/04/xmldsig-more#rsa-sha256", as shown in **3** of Example 3.8: KDM - Signature area. Any other value is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.3.10. ETM Signature Transforms Field

#### Objective

Verify that <Reference> elements of the <SignedInfo> element in the <Signature> section of the KDM do not contain a Transforms attribute.

#### Procedures

Using a text editor, view the KDM and confirm that the <Reference> elements of the <SignedInfo> element in the <Signature> section of the KDM do not contain a Transforms attribute. The presence of the Transforms attribute is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	
[SMPTE-430-3]	

Test Equipment
Text Editor

### 3.3.11. ETM Signature DigestMethod Element

#### Objective

Verify that the value of the Algorithm attribute of the <DigestMethod> element of each of the <Reference> elements in the <SignedInfo> element of the <Signature> section of the KDM is "http://www.w3.org/2001/04/xmlenc#sha256".

#### Procedures

Using a text editor, view the KDM and confirm that the value of the Algorithm attribute of the <DigestMethod> element of each of the <Reference> elements is "http://www.w3.org/2001/04/xmlenc#sha256", as shown in **5** of Example 3.8: KDM - Signature area. Any other value is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-3]	

Test Equipment
Text Editor

### 3.3.12. ETM Signature Validity

#### Objective

Verify that the signature is properly formed, i.e., the <Signature> element is properly encoded, all digests are properly formed, the <SignatureMethod> and <CanonicalizationMethod> in the <SignedInfo> element are correct, and the <Reference> values are correct. Verify that the signature is valid.

#### Procedures

Verifying that the signature is well formed (the XML structure is correct) and verifying that the signature is valid (is properly encoded) can be done by validating the signature XML against the schema using an validating XML parser, then validating the signature.

1. Using the schema validating tool **schema-check**, validate the KDM against the schema found in [SMPTE-430-3] as described in Section 1.3, i.e.,

```
$ schema-check <input-file> smpte-430-3-2007.xsd
schema validation successful
```

If the KDM is not valid or well formed, the program will report an error. A reported error is reason to fail this test.

2. Using the **checksig** software utility, verify that there is a signature included in the KDM and that it is valid. A missing or invalid signature is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-3]	

Test Equipment
Text Editor schema-check checksig dsig_cert.py

## 3.4. KDM features

### 3.4.1. KDM MessageType Element

#### Objective

Verify that the <MessageType> element of the KDM contains the string "http://www.smpte-ra.org/430-1/2005/KDM#kdm-key-type"

#### Procedures

Using a text editor, view the KDM and confirm that the <MessageType> element of the KDM contains the string "http://www.smpte-ra.org/430-1/2005/KDM#kdm-key-type" as shown in **5** of Example 3.6: KDM - AuthenticatedPublic area. Any other value in this element is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-1]	9.8

Test Equipment
Text Editor



### 3.4.2. KDM SubjectName Element

#### Objective

Verify that the Subject Name of the recipient X.509 certificate (target certificate) is identical to the value of the <SubjectName> element of the <Recipient> element of the <KDMRequiredExtensions> element in the KDM.

#### Procedures

Comparison of the Subject Name of the certificate against the content of the SubjectName element can be achieved by viewing the text version of the certificate and comparing it to the KDM element to verify they are the same.

1. Using the method described in Example 2.1: D-Cinema Certificate, view the text information of the certificate and identify the x.509 subject name as shown in [9](#).
2. Using a text editor, view the contents of the KDM and identify the <SubjectName> of the <Recipient> element as shown in [11](#).
3. Confirm that the value of the <SubjectName> element is the same as the Subject Name of the certificate. Differing values are cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	
[SMPTE-430-2]	

Test Equipment
Text Editor openssl

### 3.4.3. KDM ContentAuthenticator Element

#### Objective

Verify that, when present, the <ContentAuthenticator> element of the <KDMRequiredExtensions> element of the KDM contains one of the certificate thumbprints of one of the certificates in the chain of the signer of the CPL.

#### Procedures

If the element exists in the KDM:

1. Using a text editor, view value of the <ContentAuthenticator> element of the <KDMRequiredExtensions> element of the KDM. If the element is not present, this test is considered passed and the remaining procedure steps are not performed.
2. Extract the certificates from the KDM as described in Section 1.3.
3. Using the certificate thumbprint calculator tool **dc-thumbprint**, calculate the thumbprint each of the certificates:

```
$ dc-thumbprint <certificate.pem>
```

4. Confirm that the <ContentAuthenticator> value matches one of the thumbprints of the certificate chain of the signer certificate.

Presence of the <ContentAuthenticator> with a value that does not match one of the thumbprints is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-429-7]	
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.4.4. KDM Signer Certificate Presence

#### Objective

Verify that the certificate that signed the KDM is present in one of the <X509Data> elements of the <KeyInfo> elements in the signature portion of the KDM.

#### Procedures

Testing that the certificate that signed the KDM is present in an <X509Data> element can be achieved by validating the signature. If the validation is successful then the certificate that signed the KDM is present. The signature can be validated using the **checksig** command:

```
$ checksig <kdm-file.kdm.xml>
```

A KDM that causes **checksig** to display errors indicates that the signature did not validate and shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor checksig dsig_cert.py

### 3.4.5. KDM KeyIdList/TypedKeyId Field

#### Objective

Verify that <TypedKeyId> element of the <KeyIdList> element in the <KDMRequiredExtensions> element is well formed. Verify that the element contains one of the following values: MDIK, MDAK, MDSK, FMIK, or FMAK.

#### Procedures

To complete this test, validate the KDM against the schema in [SMPTE-430-1], then verify that one of the required values is present in the element.

1. Validate the KDM against the schema in [SMPTE-430-1] using the procedure described in Section 1.3, i.e.,

```
$ schema-check <input-file> smpte-430-1-2007.xsd
schema validation successful
```

If the KDM is not valid or well formed, the program will report an error. A reported error is cause to fail this test.

2. Verify that the <TypedKeyId> element contains one of: MDIK, MDAK, MDSK, FMIK, or FMAK, as shown in **19** of Example 3.6. Any other value in this element is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
Text Editor schema-check

### 3.4.6. KDM ForensicMarkFlagList Element

#### Objective

Verify that, if present, the <ForensicMarkFlagList> element contains a list of one or both of the following two URIs:

- <http://www.smpte-ra.org/430-1/2005/KDM#mrkflg-picture-disable>
- <http://www.smpte-ra.org/430-1/2005/KDM#mrkflg-audio-disable>

#### Procedures

Using a text editor, view the KDM and confirm the presence of the <ForensicMarkFlagList> element. The absence of the element is cause to pass this test and the remainder of this procedure can be skipped. If present, the element must contain one or both of the following URI values:

- <http://www.smpte-ra.org/430-1/2005/KDM#mrkflg-picture-disable>
- <http://www.smpte-ra.org/430-1/2005/KDM#mrkflg-audio-disable>

as shown by **21** of Example 3.6. The presence of the element with any other value, or no value, is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.4.7. KDM EncryptedData Element

#### Objective

Verify that element <EncryptedData> is not present.

#### Procedures

Using a text editor, view the KDM and confirm that the <EncryptedData> element is not present. The presence of the element is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.4.8. KeyInfo Element

#### Objective

If present, verify that the values of each <KeyInfo> element of all <EncryptedKey> elements in the <AuthenticatedPrivate> section of the KDM are identical.

#### Procedures

Using a text editor, view the KDM and, if present, confirm that the <KeyInfo> values are identical in all instances of <EncryptedKey> elements. The absence of <KeyInfo> elements is cause to pass this test. The presence of differing <KeyInfo> values in <EncryptedKey> elements is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.4.9. KDM DeviceListDescription Element

#### Objective

Verify that when present, the value of the <DeviceListDescription> element is in a human-readable language. If the optional `xml:lang` attribute is present, the language must match. If the `xml:lang` attribute is not present, the language must be English.

#### Procedures

See Objective.

Using a text editor, view the KDM and confirm that the <DeviceListDescription> element is either absent or is present and contains human-readable text. The presence of non-human-readable data or text in a language other than English without that language's corresponding `xml:lang` value is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
Text Editor



### 3.4.10. KDM ContentTitleText Language Attribute

#### Objective

Verify that value of the <ContentTitleText> element is in a human-readable language. If the optional `xml:lang` attribute is present, the language must match. If the `xml:lang` attribute is not present, the language must be English.

#### Procedures

Using a text editor, view the KDM and confirm that the <ContentTitleText> element as indicated by **13** in the Example 3.6 is a human-readable language. The presence of non-human-readable data or text in a language other than English without that language's corresponding `xml:lang` value is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor

### 3.4.11. KDM KeyType Scope Attribute

#### Objective

Verify that the optional scope attribute of the <TypedKeyId> element of the <KeyIdList> element is absent or contains the value `http://www.smpte-ra.org/430-1/2005/KDM#kdm-key-type`.

#### Procedures

Using a text editor, view the KDM and confirm that the scope attribute of the <TypedKeyId> element is either not present or is present and contains the value `http://www.smpte-ra.org/430-1/2005/KDM#kdm-key-type`, as shown in **19** of Example 3.6. Presence of the scope attribute with any other value is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	
[SMPTE-430-3]	


Test Equipment
Text Editor

### 3.4.12. KDM EncryptionMethod

#### Objective

Verify that the Algorithm attribute of the <EncryptionMethod> element of the <EncryptedKey/> element has the value "http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p".

#### Procedures

Using a text editor, view the KDM and confirm that the Algorithm attribute of the <EncryptionMethod> of the <EncryptedKey/> element contains the value http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p, as shown in  of Example 3.7: KDM - AuthenticatedPrivate area. Presence of the scope attribute with any other value is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	
[SMPTE-430-3]	

Test Equipment
Text Editor
openssl

### 3.4.13. KDM CompositionPlaylistId Element

#### Objective

Verify that the value of the <CompositionPlaylistId> element in the KDM matches the value in the RSA protected <EncryptedKey> structure, and that these values match the value of the <Id> element in the respective composition playlist.

#### Procedures

The data in the encrypted portion of the KDM can be viewed using the **kdm-decrypt** tool included in Section C.1. To view the data contained in the encrypted section of the KDM, run the command specifying the KDM and the RSA private key corresponding to the certificate to which the KDM was targeted, i.e.,

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
```

Verify that the <CompositionPlaylistId> element of the <KDMRequiredExtensions> element in the plaintext portion of the KDM contains the same value as the CPL ID present in the RSA protected <EncryptedKey> structure. Non-identical values shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-429-7]	
[SMPTE-430-1]	

Test Equipment
Text Editor kdm-decrypt

### 3.4.14. KDM Validity Fields

#### Objective

Verify that value of the <ContentKeysNotValidBefore> and <ContentKeysNotValidAfter> elements match their counterparts in the RSA protected <EncryptedKey> structure and that the values are in UTC format.

#### Procedures

The information in the encrypted portion of the KDM can be viewed using the **kdm-decrypt** tool included in Section C.1. To view the data contained in the encrypted section of the KDM, run the command specifying the KDM and the RSA private key corresponding to the certificate to which the KDM was targeted, i.e.,

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
```

Verify that the <ContentKeysNotValidBefore> element of the <KDMRequiredExtensions> element has the same value as the corresponding field inside the RSA protected EncryptedKey structure, and that it is in UTC format as specified in [RFC-3339]. Non-identical values shall be cause to fail this test.

Verify that the <ContentKeysNotValidAfter> element of the <KDMRequiredExtensions> element has the same value as the corresponding field inside the RSA protected EncryptedKey structure, is in UTC format as specified in [RFC-3339]. Non-identical values shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[RFC-3339]	
[SMPTE-430-1]	

Test Equipment
Text Editor openssl

### 3.4.15. KDM KeyIdList Element

#### Objective

Verify that each of the KeyID values in the <KeyIdList> element of the <KDMRequiredExtensions> element matches a KeyID in the RSA protected <EncryptedKey> structure and that there are no KeyIDs without corresponding <EncryptedKey> structures, nor <EncryptedKey> structures with KeyIDs that are not present in the KeyIDList.

#### Procedures

The data in the encrypted portion of the KDM can be viewed using the **kdm-decrypt** tool included in Section C.1. To view the data contained in the encrypted section of the KDM, run the command specifying the KDM and the RSA private key corresponding to the certificate to which the KDM was targeted, i.e.,

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
```

Compare the list of KeyIDs to the KeyIDs in the RSA protected EncryptedKey structures and verify that each of the KeyIDs in the list correspond to a KeyID in an RSA protected EncryptedKey structure. The presence of KeyIDs in the KeyIDList that do not correspond to a KeyID in an RSA protected EncryptedKey structure shall be cause to fail this test. The presence of a KeyID in an RSA protected EncryptedKey structure that is not also present in the KeyIDList shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
kdm-decrypt
Text Editor

### 3.4.16. KDM CipherData Structure ID

#### Objective

Verify that the value of the CipherData Structure ID in the RSA protected <EncryptedKey> structure is f1dc124460169a0e85bc300642f866ab.

#### Procedures

The data in the encrypted portion of the KDM can be viewed using the **kdm-decrypt** tool included in Section C.1. To view the data contained in the encrypted section of the KDM, run the command specifying the KDM and the RSA private key corresponding to the certificate to which the KDM was targeted, i.e.,

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
```

Verify that the plaintext value of the CipherData Structure ID is f1dc124460169a0e85bc300642f866ab. Any other value shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
kdm-decrypt

### 3.4.17. KDM CipherData Signer Thumbprint

#### Objective

Verify that the thumbprint of the signer's certificate in the RSA protected <EncryptedKey> element matches the thumbprint of the certificate that signed the KDM.

#### Procedures

The data in the encrypted portion of the KDM can be viewed using the **kdm-decrypt** tool included in Section C.1. To view the data contained in the encrypted section of the KDM, run the command specifying the KDM and the RSA private key corresponding to the certificate to which the KDM was targeted, i.e.,

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
```

A certificate thumbprint can be calculated using the **dc-thumbprint** tool included in Section C.1. Calculate the thumbprint with **dc-thumbprint**, i.e.,

```
$dc-thumbprint <certificate.pem>
```

Identify the certificate used to sign the KDM and calculate its thumbprint. Compare this thumbprint against the thumbprint decrypted from the <EncryptedKey> element and confirm that they are the same. Non-identical values shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	
[SMPTE-430-2]	

Test Equipment
dc-thumbprint
kdm-decrypt
Text Editor



### 3.4.18. KDM CipherData Validity

#### Objective

Verify that the two CipherData validity fields contain UTC format time values.

#### Procedures

The data in the encrypted portion of the KDM can be viewed using the **kdm-decrypt** tool included in Section C.1. To view the data contained in the encrypted section of the KDM, run the command specifying the KDM and the RSA private key corresponding to the certificate to which the KDM was targeted, i.e.,

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
```

Verify that the plaintext representation of the <EncryptedKey> element contains two validity time stamps in UTC format. Time stamps that are not present or that are not in UTC format shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
Text Editor
kdm-decrypt

### 3.4.19. KDM CipherData CPL ID

#### Objective

Verify that the CipherData Composition Playlist ID is identical to the value of the <CompositionPlaylistId> element in the other portions of the KDM.

#### Procedures

The data in the encrypted portion of the KDM can be viewed using the **kdm-decrypt** tool included in Section C.1. To view the data contained in the encrypted section of the KDM, run the command specifying the KDM and the RSA private key corresponding to the certificate to which the KDM was targeted, i.e.,

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
```

Verify that the decrypted plaintext value of the CompositionPlaylistID the same as the <CompositionPlaylistId> element in the AuthenticatedPublic area of the KDM. Mismatching composition playlist IDs shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	

Test Equipment
Text Editor
openssl

### 3.4.20. KDM EncryptedKey KeyType

#### Objective

Verify that the key types in the <EncryptedKey> elements of the KDM use only the allowed key types ( MDIK, MDAK, MDSK, FMIK and FMAK), and that they match the plaintext fields in the <TypedKeyId> element values for the KeyIDs in the <KeyIdList> element.

#### Procedures

The data in the encrypted portion of the KDM can be viewed using the **kdm-decrypt** tool included in Section C.1. To view the data contained in the encrypted section of the KDM, run the command specifying the KDM and the RSA private key corresponding to the certificate to which the KDM was targeted, i.e.,

```
$ kdm-decrypt <kdm-file> <rsa-private-key.pem>
```

For each <EncryptedKey> element, verify that the plaintext representation contains a key type that is one of MDIK, MDAK, MDSK, FMIK or FMAK, and that the key type is identical to the key type for the corresponding KeyID in the KeyIDList. A key type that is not either MDIK, MDAK, MDSK, FMIK or FMAK shall be cause to fail this test. A key type in the <EncryptedKey> element that does not match the key type for the corresponding KeyID in the KeyIDList shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Equipment
Text Editor kdm-decrypt

### 3.4.21. KDM Recipient X509IssuerName

#### Objective

Verify that the Distinguished Name value in the <X509IssuerName> element is compliant with [RFC-2253].

#### Procedures

Using a text editor, view the KDM and confirm that the <X509IssuerName> element as shown below **8** of Example 3.6: KDM - AuthenticatedPublic area. Verify that any special characters are properly escaped, and the sequence is correct and valid. Improperly escaped characters or sequences that do not conform to [RFC-2253] shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[RFC-2253]	
[SMPTE-430-1]	

Test Equipment
Text Editor

## 3.5. KDM Decoder Behavior

The tests in this section describe the correct behavior of a KDM decoding device.

### 3.5.1. KDM NonCriticalExtensions Element

#### Objective

Verify that a decoding device does not reject a KDM when the <NonCriticalExtensions> element is present and not empty.

#### Procedures

Perform an operation on the device under test using *KDM with non-empty NonCriticalExtensions*, a KDM that contains the <NonCriticalExtensions> with values. Verify that the operation is successful. A failed operation shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8
[SMPTE-430-1]	
[SMPTE-430-3]	

Test Material
KDM with non-empty NonCriticalExtensions

### 3.5.2. ETM IssueDate Field Check

#### Objective

Verify that the device under test verifies that the signer's certificate is valid at the time when the KDM was issued. Verify that the device under test verifies that the KDM validity does not extend beyond the ending validity period of the certificate.

#### Procedures

For each of the malformations below, perform an operation on the device under test with the test material that has that malformation. Verify that the operation fails. A successful operation is cause to fail this test.

1. KDM in which the certificate that signed the KDM has an ending validity date prior to the KDM issue date (*KDM with expired Signer certificate*).
2. KDM in which the certificate that signed the KDM has a starting validity date after to the KDM issue date (*KDM issued before certificate valid*).
3. KDM in which the validity period extends beyond the end of the signing certificate's validity period (*KDM validity exceeds signer validity*).

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.8

Test Material
KDM with expired Signer certificate
KDM issued before certificate valid

### 3.5.3. Maximum Number of DCP Keys

#### Objective

Verify that the system supports compositions with up to 256 different essence encryption keys.

#### Procedures

1. Create a show that contains the encrypted compositions *KDM for 128 Reel Composition, "A" Series* and *KDM for 128 Reel Composition, "B" Series*. Each composition contains 128 reels of encrypted picture and sound.
2. Play the show and record the following:
  - a. Whether the playout completes.
  - b. The presence of any observable artifacts in the reproduced picture and/or sound.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.7.7

Test Material
KDM for 128 Reel Composition, "A" Series
KDM for 128 Reel Composition, "B" Series

### 3.5.4. Structure ID Check

#### Objective

Verify that the device under test checks the validity of the CipherData Structure ID as specified in [SMPTE-430-1] and rejects the KDM if the Structure ID is incorrect.

#### Procedures

Perform an operation on the device under test using *KDM with corrupted CipherData block*, the KDM with an invalid CipherData Structure. Verify that the operation fails. A successful operation is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Material
KDM with corrupted CipherData block



### 3.5.5. Certificate Thumbprint Check

#### Objective

Verify that the device under test checks that the thumbprint of the signer's certificate matches the signer of the KDM and rejects the KDM if it does not.

#### Procedures

Perform an operation on the device under test using the KDM with a signer's certificate whose thumbprint does not match the thumbprint of the certificate used to sign the KDM (*KDM with incorrect signer thumbprint*). Verify that the operation fails. A successful operation is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-1]	9.8

Test Material
KDM with incorrect signer thumbprint

### 3.5.6. Certificate Presence Check

#### Objective

Verify that when the <ContentAuthenticator> element is present, the device under test checks that one of the certificates in the certificate chain supplied with the CPL has a certificate thumbprint that matches the value of <ContentAuthenticator> element.

#### Procedures

Perform an operation on the device under test using the KDM with a <ContentAuthenticator> value of a certificate thumbprint that does not match the thumbprint of one of the certificates in the certificate chain (*KDM without signer certificate*). Verify that the operation fails. A successful operation is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Material
KDM without signer certificate

### 3.5.7. KeyInfo Field Check

#### Objective

Verify that when KeyInfo elements are present in the <EncryptedKey> elements of the <AuthenticatedPrivate> area of the KDM, the device under test verifies that they all match, and that the device under test rejects the KDM if they do not match.

#### Procedures

Perform an operation on the device under test using the KDM with KeyInfo element values that do not match (*KDM with KeyInfo mismatch*). Verify that the operation fails. A successful operation is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[SMPTE-430-1]	

Test Material
KDM with KeyInfo mismatch

**Page Intentionally Left Blank**

# Chapter 4. Digital Cinema Packaging

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 2009

The DCP is the file format for d-cinema content. Entire suites of standards documents from SMPTE define this format, most notably the 428 and 429 multi-part documents. In addition, many IETF documents and some ISO documents are referenced from the SMPTE works. Reading and understanding all of these documents is a substantial task, but it is essential knowledge for accurate and efficient analysis of d-cinema files.

In the following procedures, simple tools are used to display the contents of d-cinema files. Example output from these tools is shown with descriptions of the features that will be interesting to the Test Operator. In addition to the tools used in this text, the Test Operator may use more sophisticated methods so long as the results obtained are equivalent to the procedures presented here. The reader should also note that a programmer's text editor and a binary viewer or editor are essential tools for direct inspection of data.

## 4.1. Asset Map

D-cinema track files and composition playlists are identified by unique, embedded identifiers. These identifiers, called *UUIDs*, are defined by [RFC-4122]. d-cinema XML files use UUIDs to refer to other d-cinema XML files and MXF files (assets). When d-cinema assets are written to a filesystem (*e.g.*, a disk volume), a mechanism is needed to relate the UUID values to filename values in the filesystem. An Asset Map is an XML document that provides a mapping from UUID values to filesystem paths. When a d-cinema package is written to a volume, an Asset Map is created that includes the size and location of every file in the package.<sup>1</sup> Along with the Asset Map, each volume has a Volume Index file. The Volume Index file is used to differentiate volumes in a multiple-volume distribution. Both Asset Maps and Volume Indexes are XML files (as described in Section 3.1). The formats of the Asset Map file and the Volume Index file are specified in [SMPTE 429-9].

### Example 4.1. Asset Map

```
<?xml version="1.0" encoding="UTF-8"?> 1
<AssetMap xmlns="http://www.smpite-ra.org/schemas/429-9/2007/AM"> 2
  <Id>urn:uuid:425e93f7-bca2-4255-b8ec-8c7d16fc8881</Id> 3
  <Creator> Packaging Tools v1.0 </Creator> 4
  <VolumeCount>1</VolumeCount> 5
  <IssueDate>2007-07-06T18:25:42-00:00</IssueDate>6
  <Issuer>user@host</Issuer> 7
  <AssetList> 8
    <Asset> 9
      <Id>urn:uuid:034b95b0-7424-420f-bbff-a875a79465a5</Id> 10
      <PackingList>true</PackingList> 11
      <ChunkList> 12
        <Chunk> 13
          <Path>perfect_movie_domestic_51.pk1.xml</Path> 14
          <VolumeIndex>1</VolumeIndex> 15
          <Offset>0</Offset> 16
          <Length>14366</Length> 17
        </Chunk>
      </ChunkList>
    </Asset>
  </AssetList>
  <Id>urn:uuid:4f89a209-919b-4f21-a1d6-21ad32581115</Id>
  <ChunkList>
    <Chunk>
      <Path>perfect_movie_j2c_r01.mxf</Path>
      <VolumeIndex>1</VolumeIndex>
      <Offset>0</Offset>
```

<sup>1</sup>Or packages; volumes can contain multiple DCPs.

```

        <Length>342162304</Length>
      </Chunk>
    </ChunkList>
  </Asset>
  <Asset>
    <Id>urn:uuid:e522f7b6-6731-4df5-a80e-8cfd74f82219</Id>
    <ChunkList>
      <Chunk>
        <Path>perfect_movie_wav_r01.mxf</Path>
        <VolumeIndex>1</VolumeIndex>
        <Offset>0</Offset>
        <Length>34591246</Length>
      </Chunk>
    </ChunkList>
  </Asset>

  [additional assets omitted for brevity]
  ...
</AssetList>
</AssetMap>

```

## Assetmap descriptions

- 1** XML Declaration. This specifies the version of the XML standard to which the document conforms, and the character encoding of the document.
- 2** The root Assetmap element. This element contains the XML namespace declaration for an Assetmap as specified in [SMPTE-429-9].
- 3** The Unique Universal ID (UUID) of the asset map. This is used to uniquely identify the asset map.
- 6** The date the asset map was issued.
- 7** The organization or entity that issued the asset map.
- 4** The person, software, or system that generated the asset map.
- 5** The Volume count indicates the total number of volumes that are referenced by the asset map.
- 8** The AssetList contains all of the assets in the asset map. Each asset is described in an Asset sub-element of the AssetList.
- 9** The Asset element contains all the data about an asset necessary to locate it in the filesystem.
- 10** The Asset UUID is the unique ID of a particular asset in the asset map.
- 11** The Packinglist element identifies whether or not the asset being described is a Packing List document.
- 12** The Chunklist contains the list of chunks that comprise the complete asset.
- 13** The Chunk element.
- 14** The asset chunk path is the path and filename, in the file system, of the file that contains the asset data.
- 15** The chunk volume index indicates the volume number on which the chunk resides.
- 16** The chunk offset is the number of bytes from the beginning of the complete asset file that this chunk begins. A chunk that is either a complete file or that is the beginning of a file will have an offset of 0.
- 17** The chunk length is the length, in bytes, of the chunk of the asset.

### Example 4.2. Volume Index

```

<?xml version="1.0" encoding="UTF-8"?> 1
<VolumeIndex xmlns="http://www.smpte-ra.org/schemas/429-9/2007/AM"> 2
  <Index>1</Index> 3
</VolumeIndex>

```

## Volume Index descriptions

- 1** XML Declaration. This specifies the version of the XML standard to which the document conforms, and the character encoding of the document.
- 2** The root Assetmap element. This element contains the XML namespace declaration for an Assetmap as specified in [SMPTE-429-9].
- 3** The index number of the volume.

## 4.1.1. Asset Map File

### Objective

Verify that the Asset Map file is in the root of the volume, and that it is named ASSETMAP.xml. Verify that the Asset Map validates against the schema defined in [SMPTE-429-9-2007].

### Procedures

1. Mount the media that contains the volume with a computer, and obtain a directory listing of the root of the filesystem. The absence of the file ASSETMAP.xml is cause to fail this test.
2. Using the **schema-check** software utility, validate the file ASSETMAP.xml against the schema in [SMPTE-429-9-2007]. Failure to correctly validate is cause to fail this test. For more information on schema validation see Section 1.3: Conventions and Practices

E.g.:

```
$ cd /
$ ls -F
ASSETMAP.xml
PKL_c2434860-7dab-da2b-c39f-5df000eb2335.xml
J2K_a13c59ec-f720-1d1f-b78f-9bdea4968c7d_video.mxf
WAV_22d190bd-f43b-a420-a12e-2bf29a737521_audio.mxf
...
$
$ schema-check ASSETMAP.xml smpte-429-9-2007.xsd
schema validation successful
$
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.5.2.1
[SMPTE-429-9-2007]	

Test Equipment
schema-check

## 4.1.2. Volume Index File

### Objective

Verify that the Volume Index file is in the root of the volume and that it is named `VOLINDEX.xml`. Verify that the Volume Index file validates against the schema defined in [SMPTE-429-9].

### Procedures

1. Mount the media that contains the volume with a computer, and obtain a directory listing of the root of the filesystem. The absence of the file `VOLINDEX.xml` is cause to fail this test.
2. Using the **schema-check** software utility, validate the file `VOLINDEX.xml` against the schema in [SMPTE-429-9-2007]. Failure to correctly validate is cause to fail this test. For more information on schema validation see Section 1.3: Conventions and Practices

E.g.:

```
$ cd /
$ ls -F
VOLINDEX.xml
PKL_c2434860-7dab-da2b-c39f-5df000eb2335.xml
J2K_a13c59ec-f720-1d1f-b78f-9bdea4968c7d_video.mxf
WAV_22d190bd-f43b-a420-a12e-2bf29a737521_audio.mxf
...
$
$ schema-check VOLINDEX.xml smpte-429-9-2007.xsd
schema validation successful
$
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.5.2.1

Test Equipment
schema-check



## 4.2. Packing List

The Packing List (PKL) is an XML document (see Section 3.1) that specifies the contents of a d-cinema Package. It contains the UUID, file type (MXF track file, CPL, etc.), and a message digest of each file in the DCP. This information is used to ensure that all of the expected files have been included and have not been modified or corrupted in transit. The format of the Packing List file is specified by [SMPTE-429-8].

### Example 4.3. Packing List

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?> ❶
<PackingList xmlns="http://www.smpte-ra.org/schemas/429-8/2007/PKL"> ❷
  <Id>urn:uuid:59430cd7-882d-48e8-a026-aef4b6253dfc</Id> ❸
  <AnnotationText>Perfect Movie DCP</AnnotationText> ❹
  <IssueDate>2007-07-25T18:21:31-00:00</IssueDate> ❺
  <Issuer>user@host</Issuer> ❻
  <Creator>Packaging Tools v1.0</Creator> ❼
  <AssetList> ❽
    <Asset> ❾
      <Id>urn:uuid:24d73510-3481-4ae5-b8a5-30d9eeced9c1</Id> ❿
      <Hash>AXufMKY7NyZcfSXQ9sCZ1s5dSyE=</Hash> ⓫
      <Size>32239753</Size> ⓬
      <Type>application/mxf</Type> ⓭
    </Asset>
    <Asset>
      <Id>urn:uuid:456e547d-af92-4abc-baf3-c4d730bbcd65</Id>
      <Hash>kAAo0kXYVDBJUpHID89zauv50w=</Hash>
      <Size>86474446</Size>
      <Type>application/mxf</Type>
    </Asset>
    <Asset>
      <Id>urn:uuid:e4a4e438-63ec-46cb-b9aa-43acee787d79</Id>
      <Hash>kt5bP8y4zmHNAY1qVnujItAb4sY=</Hash>
      <Size>12163</Size>
      <Type>text/xml</Type>
    </Asset>
    <Asset>
      <Id>urn:uuid:3d445456-54d5-42bc-a7cc-a8c00b20ffb7</Id>
      <Hash>AQWMKCxxMv001zTS3Y30j8M+d9s=</Hash>
      <Size>62500144</Size>
      <Type>application/mxf</Type>
    </Asset>

    [Remaining assets and signature omitted for brevity]
  </AssetList>
  [Signature omitted for brevity]
</PackingList>
```

### Packing List descriptions

- ❶ XML Declaration. This specifies the version of the XML standard to which the document conforms.
- ❷ The root packing list element. This element contains the XML namespace declaration for the packing list as specified in [SMPTE-429-8].
- ❸ The Unique Universal ID (UUID) of the packing list.
- ❹ The date the packing list was issued.
- ❺ The organization or entity that issued the packing list.
- ❻ The person, software, or system that generated the packing list.

- 4** The Annotation text is a plain text, human readable language description of the packing list's contents.
- 8** The assetlist contains all of the assets in the packing list.
- 9** The Asset element contains all the metadata necessary to identify the file.
- 10** The Asset UUID is the unique ID of a particular asset in the packing list.
- 11** The asset hash is a message digest of the asset file.
- 12** The asset size is the size, in bytes, of the asset's file in the filesystem.
- 13** The asset type contains the mime type of the asset, which is a generic description of the file format. It also contains an attribute that specifies the specific kind of type, such as a CPL, Picture, or Sound file.

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8, 2009

## 4.2.1. Packing List File

### Objective

- Verify that the Packing List is an XML document and that it validates against the schema defined in [SMPTE-429-8-2007].
- Confirm that if the language attribute of the <AnnotationText> element is not present, or present with a value of "en", that the Annotation text is in human-readable English.
- Verify that the Packing List contains urn:uuid values as specified in [RFC-4122].
- Verify that the listed file sizes match those for each of the referenced assets.

### Procedures

In the following procedures, the callout numbers refer to Example 4.3: Packing List

1. Using the **schema-check** software utility, validate the XML file structure against the schema in [SMPTE-429-8-2007]. Failure to correctly validate is cause to fail this test. For more information on schema validation see Section 1.3: Conventions and Practices.

```
$ schema_check.py <input-file> smpte-429-8-2007.xsd
schema validation successful
$
```

2. Open the Packing List file in a text editor and verify that if the "language" attribute of the <AnnotationText> **4** element is not present, or present with a value of "en", that the contents of the <AnnotationText> **4** element is human readable English. Failure to meet this requirement is cause to fail this test.

```
$ vi <input-file>
...
  <AnnotationText>Perfect Movie Reel #1 Picture</AnnotationText>
...
  <AnnotationText language="en">Perfect Movie Reel #1 Sound</AnnotationText>
...
:q
$
```

3. Supply the filename of the Packing List file as an argument to the **uuid\_check.py** software utility. Examine the output for error messages that identify expected UUID values that do not conform to the format specified in [RFC-4122]. One or more occurrences is cause to fail this test.

Version  
NEWER  
Versi

```
$ uuid_check.py <input-file>
all UUIDs conform to RFC-4122
$
```

4. To verify that the real file sizes of the referenced assets are equal to the values of the related XML elements, the path to those assets must be known. The following procedure may be used if the ASSETMAP .xml file is available otherwise the tester will need to devise a method for locating the relevant assets. For each of the <Asset> 9 elements contained in the Packing List, compare the contents of the child <Id> 10 element with the contents of the ASSETMAP .xml file to discover the path to the asset. List the file size of the referenced asset and verify that it is identical to the value of the child <Size> 12 element inside the <Asset> 9 element. One or more failures to verify the file sizes is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.5.3.1, 5.5.3.2
[SMPTE-429-8-2007]	

Test Equipment
schema-check
uuid_check.py
Text Editor

## 4.2.2. Packing List Signature Validation

### Objective

Verify that the Packing List is signed and that the signature validates.

### Procedures

Using the **checksig** software utility, verify that there is a signature included in the Packing List and that it is valid. If the signature is missing, or invalid, this is cause to fail this test.

```
$checksig <input-file>  
The supplied signature is valid  
$
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.5.2.3, 5.5.3.2
[PKCS-1]	
[RFC-3174]	
[SMPTE-429-8-2007]	

Test Equipment
checksig
dsig_cert.py

## 4.3. Composition Playlist

The Composition Playlist (CPL) is an XML document (see Section 3.1) that contains the information necessary to reproduce a composition. It contains metadata about the composition such as the title and the rating, and references to the track files that contain the composition's essence. The format of the Composition Playlist file is specified by [SMPTE-429-7].

### Example 4.4. Composition Playlist

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?> 1
  <CompositionPlaylist xmlns="http://www.smpte-ra.org/schemas/429-7/2006/CPL"> 2
    <Id>urn:uuid:20670ba3-d4c7-4539-ac3e-71e874d4d7d1</Id> 3
    <IssueDate>2007-07-25T00:35:03-00:00</IssueDate> 4
    <Issuer>user@host</Issuer> 5
    <Creator> Packaging Tools v1.0 </Creator> 6
    <ContentTitleText>Perfect Movie</ContentTitleText> 7
    <ContentKind>feature</ContentKind> 8
    <ContentVersion> 9
      <Id>urn:uuid:e5alb4dc-faf3-461b-a5e2-9d33088b1b28</Id> 10
      <LabelText>Perfect Movie - Domestic - US 5.1 </LabelText> 11
    </ContentVersion>
    <RatingList/> 12
    <ReelList> 13
      <Reel> 14
        <Id>urn:uuid:f62cffe9-2da7-4d28-b73e-f21c816ab02f</Id> 15
        <AssetList> 16
          <MainPicture>17
            <Id>urn:uuid:93270dd0-8675-42fa-9ce8-34b61c963997</Id>18
            <EditRate>24 1</EditRate>19
            <IntrinsicDuration>480</IntrinsicDuration>20
            <EntryPoint>0</EntryPoint>21
            <Duration>480</Duration>22
            <FrameRate>24 1</FrameRate>23
            <ScreenAspectRatio>1998 1080</ScreenAspectRatio>24
          </MainPicture>25
          <MainSound>26
            <Id>urn:uuid:e33b7b37-da90-4429-88af-5c5b63506017</Id>
            <EditRate>24 1</EditRate>
            <IntrinsicDuration>2880</IntrinsicDuration>
            <EntryPoint>120</EntryPoint>
            <Duration>2760</Duration>
          </MainSound>
        </AssetList>
      </Reel>
    </ReelList>
    [Additional reel data and CPL Signature omitted for brevity]
  </CompositionPlaylist>
```

### Composition Playlist descriptions

- 1 The XML version of the XML standard to which the document conforms, the character encoding of the document, and whether the document relies on external declarations or parameter entities.
- 2 The Root Composition Playlist element. This element contains the XML namespace declaration for the Composition Playlist as specified in [SMPTE-428-7].
- 3 The Unique Universal ID (UUID) of the composition playlist.
- 4 The date the CPL was issued.
- 5 The organization or entity that issued the CPL.
- 6 The person, software, or system that generated the CPL.
- 7 A descriptive string that describes the composition and is displayed to the user.

- Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released March 2009
- 8 The kind of presentation the CPL represents, such as a feature, trailer, or advertisement.
  - 9 The version of the content represented by the composition playlist. This element contains sub-elements that contain a descriptive label and UUID of the content.
  - 10 The unique ID of the version of the content represented by the CPL (as opposed to the unique ID of the CPL).
  - 11 A text description of the version of the content represented in the CPL.
  - 12 The list of ratings applied to the content represented by the CPL. In compositions that contain rating information, the <RatingList> element contains at least one instance of the <Rating> element, which in turn contains two elements, <Agency>, that contains a URI that represents the agency that issued the rating, and <Label>, that contains the rating.
  - 13 The list of reels that comprise the composition.
  - 14 A reel of the composition.
  - 15 The unique ID of the reel.
  - 16 The list of assets that comprise the reel.
  - 17 The element in the reel that contains the information required to produce images onscreen.
  - 18 The unique ID of the MXF track file that contains the picture essence (the picture track file) to be reproduced onscreen.
  - 19 The edit rate, or the number of editable units of content, per second, of the picture track file.
  - 20 The total number of frames in the track file, inclusive of frames not intended for reproduction onscreen.
  - 21 The first frame of the track file to be reproduced onscreen.
  - 22 The number of frames of the track file to be reproduced onscreen. When a picture track file is present in a composition, its duration is effectively the duration of the reel.
  - 23 The rate, in frames-per-second, at which the essence in the track file will be reproduced.
  - 24 The aspect ratio of the essence in the picture track file. This is represented in the CPL as a ratio of two numbers separated by a space.
  - 26 The element in the reel that contains the information required to reproduce sound essence through the primary speaker system. The parameters of a MainSound track file are the same as those of a picture track file.

### 4.3.1. Composition Playlist File

#### Objective

Verify that the Composition Playlist is an XML document and that it validates against the schema defined in [SMPTE-429-7].

#### Procedures

Using the **schema-check** software utility, validate the XML file structure against the schema in [SMPTE-429-7]. Failure to correctly validate is cause to fail this test.

```
$ schema-check <input-file> smpte-429-7.xsd
schema validation successful
$
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.2.3, 5.4.2, 5.4.3, 9.7.7, 5.4.3.2, 5.4.3.3, 5.4.3.4
[SMPTE-429-7]	

<b>Test Equipment</b>
schema-check

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Discontinued May 8 2009

## 4.3.2. Composition Playlist Signature Validation

### Objective

Verify that the Composition Playlist is signed and that the signature validates.

### Procedures

Using the **checksig** software utility, verify that there is a signature included in the Composition Playlist List and that it is valid. If the signature is missing, or invalid, this is cause to fail this test.

Example:

```
$checksig <input-file>  
The supplied signature is valid  
$
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.2.3, 5.4.3.6, 5.4.4, 9.7.5

Test Equipment
dsig_cert.py checksig



### 4.3.3. Track File Key Epoch

#### Objective

An encrypted Asset is associated with a Decryption Key that is effective for a period of time equal to one Reel. Only one Decryption Key shall be associated with a specific encrypted Asset. Each unique Decryption Key shall be associated with only one encrypted Asset.

- Verify that for each encrypted Asset present in the Composition Playlist, only one <KeyId> value is listed. If an Asset Id occurs more than once in the CPL, verify that the same <KeyId> is utilized throughout.
- Verify that each <KeyId> is associated with only one Asset Id.

#### Procedures

1. Use a text editor to view the Composition Playlist. For all encrypted Assets (those that have a <KeyId> value) make a list of all Asset Id values and the associated <KeyId> values.
2. Examine the list to determine that each Asset Id has exactly one <KeyId>. If Asset Ids are repeated in the CPL, the same <KeyId> should be associated for that Asset every time. Any deviation is cause to fail this test.
3. Examine the list to determine that each <KeyId> is associated with exactly one Asset Id (i.e. a particular Decryption Key should only be associated with one, unique Asset). Any deviation is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.3.1.7

Test Equipment
Text Editor

## 4.4. Track Files

A Track File is a container for encoded essence. In the d-cinema system, each Track File contains a single track of a single type of essence<sup>2</sup>. For example, a Track File may contain images or sound or timed text, but never more than one type of essence.

D-cinema Track Files are based on the Material eXchange Format (MXF). MXF is a file metaformat, *i.e.*, a file format for creating file formats. While the various d-cinema Track File formats represent different methods of encoding essence data, the arrangement of metadata within the files is syntactically similar. This section will provide an overview of MXF as used for d-cinema applications. Readers looking for more detailed technical information are referred to [SMPTE-EG41] and [SMPTE-EG42].

### 4.4.1. MXF Internals

#### 4.4.1.1. Overview

Before diving head-first into examining MXF files, it is important to understand the structure of the files. This section will briefly describe the contents of some example MXF files by displaying the files' header metadata using the **klvwalk** software utility from the free ASDCPLib software package.

Briefly, an MXF file [SMPTE 377M-2004] contains a sequence of Key-Length-Value (KLV) packets. Some packets carry essence and some carry metadata. MXF files are divided into *partitions*. Each partition is comprised of a set of KLV packets. The first KLV packet in each partition is a Partition Pack.

The number of partitions in a digital cinema sound or picture Track File is usually three (Timed Text Track Files may have more than three partitions). The first partition in an MXF file contains the metadata which describe the coding parameters of the essence and the MXF file itself. The second partition contains the essence data as a sequence of KLV-wrapped frames. The final partition contains the index table.

To display the metadata in the header partition of an MXF file `testfile.mxf`, use **klvwalk** like so:

```
$ klvwalk -r testfile.mxf
...
```

The following sections illustrate the expected output.

#### 4.4.1.2. MXF Header Partition

As shown in Example 4.5, the first structure to be output is the Partition Pack of the Header Partition. This structure documents the MXF version that the file conforms to and provides a description of the general architecture to be found inside.

##### Example 4.5. MXF Partition Header

```
06.0e.2b.34.02.05.01.01.0d.01.02.01.01.02.04.00 len: 120 (ClosedCompleteHeader) 1
```

<sup>2</sup> Strictly speaking, a Timed Text Track File may contain font and image resources in addition to the XML timed text data, but these resources are considered integral to the timed text essence.

```

MajorVersion      = 1
MinorVersion      = 2
KAGSize           = 1
ThisPartition     = 0
PreviousPartition = 0
FooterPartition   = 218362864
HeaderByteCount   = 16244
IndexByteCount    = 0
IndexSID          = 0
BodyOffset        = 0
BodySID           = 1
OperationalPattern = 060e2b34.0401.0101.0d010201.10000000 2
Essence Containers: 3
060e2b34.0401.0103.0d010301.027f0100
060e2b34.0401.0107.0d010301.020b0100

```

### MXF Partition Header

- 1 This is an MXF Partition Pack structure. The Universal Label (UL) value indicates that the file is "Closed and Complete".
- 2 The Operational Pattern UL indicates that the file conforms to OP Atom [SMPTE 390M].
- 3 Essence Container labels indicate the type of essence and the wrapping format. This example shows two container labels: the JPEG 2000 container [SMPTE-422M] and the Generic Container [SMPTE-379M] (the file contains encrypted JPEG 2000 essence).

The following table gives the list of valid Essence Container ULs for d-cinema Track Files.

**Table 4.1. Essence Container UL Values for d-cinema**

UL Value	Container Type
060e2b34.0401.0101.0d010301.02060100	Linear PCM Audio [SMPTE-429-3], [SMPTE-382M]
060e2b34.0401.0107.0d010301.020c0100	JPEG 2000 Images [SMPTE-429-4]
**Not yet assigned**	Timed Text [SMPTE-429-5]
060e2b34.0204.0101.0d010301.027e0100	Encrypted Essence [SMPTE-429-6]

### 4.4.1.3. File Package

An MXF file may contain zero or more continuous segments of essence data. Each segment is described by a Source Package structure. Per [SMPTE-429-3-2006], MXF files for digital cinema must contain exactly one top-level Source Package (thus one segment of essence), referred to in MXF jargon as a File Package. Example 4.6 shows a Source Package structure that points to JPEG 2000 essence data.

#### Example 4.6. Source Package structure

```

06.0e.2b.34.02.53.01.01.0d.01.01.01.01.01.37.00 len:      294 (SourcePackage) 1
  InstanceUID = 42b5a376-c740-42e2-99f1-4ec782c4837e
  PackageUID = [060a2b34.0101.0105.01010f20],13,00,00,00,
               [b4f492cd.b89b.0f65.490c35ec.5f6340b7] 2
  Name = File Package: SMPTE 429-4 frame wrapping of JPEG 2000 codestreams
  PackageCreationDate = 2007-03-21 07:42:04.000
  PackageModifiedDate = 2007-03-21 07:42:04.000
  Tracks: 3
9227a330-7e64-4c90-b4ef-d057ed6ef159
0de983e3-255b-4d26-bde7-f33c530c077d
54e13d93-abcfc-4869-b008-c59573b8d01d
  Descriptor = c6a35640-d6d8-433c-82c9-23df2eae9311 4

```

## Source Package structure

- 1 This is a Source Package structure [SMPTE-377M].
- 2 A Unique Material Identifier (UMID) value which identifies the essence in the file. It has a UUID component which is the value that external entities (*e.g.* Packing Lists and Composition Playlists) use to refer to the essence in the file. See [SMPTE 429-3] for details about how d-cinema UMIDs are formed.
- 3 The list of tracks that appear in the file. There is only one essence track, but it is accompanied by a virtual timecode track and, optionally, a descriptive metadata track that gives cryptographic information (see Section 4.4.1.4 below).
- 4 This value gives the internal ID of a data set that describes the essence encoding. This set is called an Essence Descriptor. Two examples of essence descriptors are given below in Section 4.4.1.5 and Section 4.4.1.6.

### 4.4.1.4. Encrypted Essence

If the MXF file contains encrypted essence, the header metadata will contain one Cryptographic Framework set with a link to a single Cryptographic Context set (defined in [SMPTE-429-6-2006]). These structures are shown in Example 4.7.

#### Example 4.7. Cryptographic Framework and Cryptographic Context

```
06.0e.2b.34.02.53.01.01.0d.01.04.01.02.01.00.00 len:      40 (CryptographicFramework) 1
    InstanceUID = b98ca683-2e49-4e6a-88ff-af33910ba334
    ContextSR = 8dcd2f7b-fd0b-4602-bae7-806c82dcfd94

06.0e.2b.34.02.53.01.01.0d.01.04.01.02.02.00.00 len:      120 (CryptographicContext) 2
    InstanceUID = 8dcd2f7b-fd0b-4602-bae7-806c82dcfd94
    ContextID = 3472d593-e9ff-4b2e-84ca-5303b5ce53f7
    SourceEssenceContainer = 060e2b34.0401.0107.0d010301.020c0100 3
    CipherAlgorithm = 060e2b34.0401.0107.02090201.01000000 4
    MICAlgorithm = 060e2b34.0401.0107.02090202.01000000 5
    CryptographicKeyID = c030f37a-bf84-496b-bdc2-81744205a944 6
```

### Cryptographic Framework and Cryptographic Context

- 1 This is a Cryptographic Framework structure [SMPTE-429-6-2006].
- 2 This is a Cryptographic Context structure [SMPTE-429-6-2006].
- 3 A UL that identifies the type of essence inside the encrypted container. It should be a JPEG 2000 or PCM audio descriptor.
- 4 A UL that identifies the type of encryption used. This value should always be 060e2b34.0401.0107.02090201.01000000.
- 5 A UL that identifies the algorithm used to calculate the Message Integrity Check value in each Encrypted KLV (EKL) packet. When present, this value should always be 060e2b34.0401.0107.02090202.01000000.
- 6 A UUID value that identifies the 16-byte symmetric key (stored externally) that is required to decrypt the essence data. The key is usually delivered to a system via a Key Delivery Message (see Chapter 3).

### 4.4.1.5. Essence Descriptor for JPEG 2000

If the MXF file contains image essence for DCI-compliant digital cinema, the header metadata will contain an RGBA Essence Descriptor (defined in [SMPTE-377M-2004], with a strong link to a JPEG 2000 Picture SubDescriptor (defined in [SMPTE-422M]). These structures are shown in Example 4.8

#### Example 4.8. Essence Descriptor for JPEG 2000

```

06.0e.2b.34.02.53.01.01.0d.01.01.01.01.01.29.00 len: 169 (RGBAEssenceDescriptor) 1
    InstanceUID = 18a47da5-53d1-4785-a91e-41155753a02f
    Locators:
    SubDescriptors:
    05f80258-beb2-4769-b99a-af4d6c3895da
        LinkedTrackID = 2
        SampleRate = 24/1 2
    ContainerDuration = 720 3
    EssenceContainer = 060e2b34.0401.0107.0d010301.020c0100
        Codec = 060e2b34.0401.0109.04010202.03010103 4
        FrameLayout = 0
        StoredWidth = 2048 5
        StoredHeight = 1080 6
        AspectRatio = 2048/1080
        ComponentMaxRef = 4095
        ComponentMinRef = 0

06.0e.2b.34.02.53.01.01.0d.01.01.01.01.01.5a.00 len: 174 (JPEG2000PictureSubDescriptor) 7
    InstanceUID = 05f80258-beb2-4769-b99a-af4d6c3895da
        Rsize = 3
        Xsize = 2048
        Ysize = 1080
        XOsize = 0
        YOsize = 0
        XTsize = 2048
        YTsize = 1080
        XTOsize = 0
        YTOsize = 0
        Csize = 3
    PictureComponentSizing = 00000003000000030b01010b01010b0101
    CodingStyleDefault = 0104000101050303000077888888888888
    QuantizationDefault = 227f187f007f007ebc76ea76ea76bc6f4c6f4c6f645803580358455fd25fd25f61

```

### Essence Descriptor for component images

- 1 This is an MXF RGBA Essence Descriptor structure.
- 2 The frame rate of the underlying essence. The essence may be sampled on a finer scale, but this value is the smallest temporal increment than can be accessed in the file.
- 3 The number of frames in the file. Divide this value by the SampleRate to get the duration as a time value in seconds.
- 4 This UL value indicates the type of compression and the color space of the encoded essence.
- 5 The width of the encoded image as a count of pixels.
- 6 The height of the encoded image as a count of pixels.
- 7 This is an MXF JPEG 2000 Picture SubDescriptor structure. It provides additional metadata associated with the JPEG 2000 encoding.

#### 4.4.1.6. Essence Descriptor for PCM Audio

If the MXF file contains audio essence for DCI-compliant digital cinema, the header metadata will contain a Wave Audio Descriptor (defined in [SMPTE-382M]). This structure is shown in Example 4.9.

#### Example 4.9. Essence Descriptor for PCM Audio

```

06.0e.2b.34.02.53.01.01.0d.01.01.01.01.01.48.00 len: 134 (WaveAudioDescriptor) 1
    InstanceUID = 0b7eac6c-85e2-47e4-b0bf-b3e60f6e6cd7
    Locators:
    SubDescriptors:
    LinkedTrackID = 2
    SampleRate = 24/1 2
    ContainerDuration = 528 3

```

```

EssenceContainer = 060e2b34.0401.0101.0d010301.02060100
AudioSamplingRate = 48000/1 4
    Locked = 0
    AudioRefLevel = 0
    ChannelCount = 6 5
QuantizationBits = 24 6
    DialNorm = 0
    BlockAlign = 18 7
    SequenceOffset = 0
    AvgBps = 144000

```

### Essence Descriptor for PCM Audio

- 1** This is a Wave Audio Descriptor structure [SMPTE-382M].
- 2** The frame rate of the underlying essence. The essence may be sampled on a finer scale, but this value is the smallest temporal increment than can be accessed in the file.
- 3** The number of frames in the file. Divide this value by the SampleRate to get the duration as a time value in seconds.
- 4** The base sample rate of the essence.
- 5** The number of channels in the file. Each frame of essence will have the same number of channels, multiplexed in the same order.
- 6** The number of bits used to encode a sample of a single channel.
- 7** The size, in bytes, of a set of samples for all channels in a single sample period. This value should be equal to  $(QuantizationBits / 8) * ChannelCount$ .

#### 4.4.1.7. MXF Header Partition

All d-cinema Track Files end with a Random Index Pack (RIP). The RIP provides a lookup table that gives the location of all partitions in the file for easy random access. The number of partitions shown by the RIP should be three if the MXF file is a sound or picture Track File, and may be more than three for a Timed Text Track File.

#### Example 4.10. MXF Random Index Pack (RIP)

```

06.0e.2b.34.02.05.01.01.0d.01.02.01.01.11.01.00 len: 40 (RandomIndexMetadata) 1
 0 : 0
 1 : 16384
 0 : 110688380

```

#### MXF Random Index Pack (RIP)

- 1** The Random Index Pack (RIP) maps the location of each partition in an MXF file. This example shows three partitions.

## 4.4.2. Image and Audio Packaging Standard

### Objective

- Verify that sound and image essence are wrapped in files conforming to Material Exchange Format (MXF) as defined by [SMPTE-377M-2004], and further constrained by [SMPTE-379M-2004], [SMPTE-429-3], and [SMPTE-429-4], [SMPTE-422M-2006] for image, or [SMPTE-382M] for sound.
- If the Essence Container is encrypted, verify that this conforms to [SMPTE-429-6].

### Procedures

1. Using the **klvwalk** software utility, produce a listing of the MXF KLV Header Metadata Structure. Error free completion of the command confirms the validity of the MXF structure. Any other result is cause to fail the test.
2. Examine the listing for the MXF Partition Pack structure with a ClosedCompleteHeader Universal Label (UL) value:  
060e2b34.0205.0101.0d010201.01020400  
as shown in Example 4.5: MXF Partition Header item **1**. Absence of this value is cause to fail this test.
3. Examine the listing for the OperationalPattern value:  
060e2b34.0401.0101.0d010201.10000000,  
as shown in Example 4.5 item **2**. Absence of this value is cause to fail this test.
4. Examine the listing for the Essence Container values as shown in Example 4.5 item **3**. There are three valid possibilities for the data in this field:
  - a. If two values are present, and they are:  
060e2b34.0401.0103.0d010301.027f0100 and  
060e2b34.0401.0107.0d010301.020c0100,  
then the file is an Image file. For more information see Section 4.4.1.5: Essence Descriptor for JPEG 2000.
  - b. If two values are present, and they are:  
060e2b34.0401.0103.0d010301.027f0100 and  
060e2b34.0401.0101.0d010301.02060100,  
then the file is an Sound file. For more information see Section 4.4.1.6: Essence Descriptor for PCM Audio.
  - c. If there is a single value:  
060e2b34.0401.0107.0d010301.020b0100,  
the Essence is Cyphertext and an additional procedure, listed below, must be carried out.  
Failure to meet exactly one of the valid possibilities is cause to fail this test.
5. Examine the listing and locate the EssenceContainerData set, UL value:  
060e2b34.0253.0101.0d010101.01012300.  
This should contain exactly one LinkedPackageUID value. Verify that there is only one SourcePackage set, UL value:  
060e2b34.0253.0101.0d010101.01013700  
and that the PackageUID value exactly matches the LinkedPackageUID value of the EssenceContainerData set.  
Failure of any of the above conditions is cause to fail this test.
6. Only for the case of Encrypted Essence, the SourcePackage set, UL value:  
060e2b34.0253.0101.0d010101.01013700,

should contain a third Track UID that matches the InstanceUID value of a single StaticTrack set, UL value:

060e2b34.0253.0101.0d010101.01013a00.

The StaticTrack set should have a Sequence value that matches the InstanceUID of a Sequence set, UL value:

060e2b34.0253.0101.0d010101.01010f00.

The found Sequence set should have a StructuralComponents value that matches the InstanceUID of a single DMSegegment set, UL value:

060e2b34.0253.0101.0d010101.01014100.

The DMSegegment set should have a DMFramework value that matches a single CryptographicFramework set, UL value:

060e2b34.0253.0101.0d010401.02010000.

The CryptographicFramework set should have a ContextSR value that matches the InstanceUID of a single CryptographicContext set, UL value:

060e2b34.0253.0101.0d010401.02020000.

The CryptographicContext set has a SourceEssenceContainer value, which should contain either the UL value:

060e2b34.0401.0107.0d010301.020c0100

for an Image file, or:

060e2b34.0401.0101.0d010301.02060100

for a Sound file. For more information see Section 4.4.1.4: Encrypted Essence. Failure of any of the above conditions is cause to fail this test.

## Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.2.2.2, 5.2.2.3, 5.2.2.4, 5.2.2.5, 5.2.2.6, 5.3.1.2, 5.3.1.10, 5.3.1.11, 5.3.1.12
[SMPTE-377M-2004]	
[SMPTE-379M-2004]	
[SMPTE-382M]	
[SMPTE-422M-2006]	
[SMPTE-429-3]	
[SMPTE-429-4]	
[SMPTE-429-6]	

Test Equipment
klvwalk



### 4.4.3. Timed Text Track File Format

#### Objective

- Verify that timed text essence is wrapped in files conforming to Material Exchange Format (MXF) as defined by [SMPTE-377M-2004] and [SMPTE-410M], and further constrained by [SMPTE-379M-2004] and [SMPTE-429-5].
- Verify that timed text essence is encoded according to [SMPTE-428-7].
- If the Essence Container is encrypted, verify that this conforms to [SMPTE-429-6].

#### Procedures

1. Using the **klvwalk** software utility, produce a listing of the MXF KLV Header Metadata structure. Error free completion of the command confirms the validity of the MXF structure. Any other result is cause to fail the test.
2. Examine the listing for the MXF Partition Pack structure with a ClosedCompleteHeader Universal Label (UL) value:  
060e2b34.0205.0101.0d010201.01020400  
as shown in Example 4.5: MXF Partition Header item **1**. Absence of this value is cause to fail this test.
3. Examine the listing for the OperationalPattern value:  
060e2b34.0401.0101.0d010201.10000000,  
as shown in Example 4.5 item **2**. Absence of this value is cause to fail this test.
4. Examine the listing for the Essence Container values as shown in Example 4.5 item **3**. There are two valid possibilities for the data in this field:
  - a. If two values are present, and they are:  
060e2b34.0401.0103.0d010301.027f0100 and  
\*\*UL value TBD\*\*,  
then the file is a Timed Text file. For more information see Section 4.4.1.5: Essence Descriptor for JPEG 2000.
  - b. If there is a single value:  
060e2b34.0401.0107.0d010301.020b0100,  
the Essence is Cyphertext and an additional procedure, listed below, must be carried out.  
Failure to meet exactly one of the valid possibilities is cause to fail this test.
5. Examine the listing and locate the EssenceContainerData set, UL value:  
060e2b34.0253.0101.0d010101.01012300.  
This should contain exactly one LinkedPackageUID value. Verify that there is only one SourcePackage set, UL value:  
060e2b34.0253.0101.0d010101.01013700  
and that the PackageUID value exactly matches the LinkedPackageUID value of the EssenceContainerData set.  
Failure of any of the above conditions is cause to fail this test.
6. Only for the case of Encrypted Essence, execute sub-procedure #6 as given in Section 4.4.2.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	3.3.4

Version  
NEWER  
Version  
0 ARCHIVE ONLY  
VERSION AVAILABLE  
Used May 8 2009

Reference Document ID	Reference Document Section(s)
[SMPTE-428-7]	
[SMPTE-429-5]	
[SMPTE-429-6]	

Test Equipment
klvwalk

## 4.4.4. Track File Length

### Objective

For each Track File, verify that the minimum duration is a number of frames which is greater or equal to one second of content playback at the specified edit rate. This means that each image Track File needs to contain at least 24 (at 24 fps frame rate) or 48 (at 48 fps frame rate) frames, and that each audio Track File needs to contain at least 48,000 (at 48kHz sampling rate) or 96,000 (at 96 kHz sampling rate) audio samples.

### Procedures

This may be accomplished by using the **asdcp-test** software utility to provide information about the file and confirming that the reported ContainerDuration value is equal or greater than the SampleRate value. Failure to meet the above conditions is cause to fail this test.

E.g.

```
$ asdcp-test -i -v <input-file>
...
SampleRate: 24/1
...
ContainerDuration: 528
...
$
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.3.1.3

Test Equipment
klvwalk

## 4.4.5. Image Track File Frame Boundary

### Objective

- Image Track Files must begin and end with complete frames that allow for splicing. Verify that both the first and the last JPEG2000 image in a sequence are completely contained within the Image Track File, i.e., no other Track Files are needed for complete decoding or displaying of the first and the last frame.
- Each complete Frame of Image Data must be wrapped within the KLV structure according to [SMPTE 336M] and [SMPTE-422M-2006].

### Procedures

1. Determine the number of frames contained in the Track File. This will be used in the next step to extract the last frame in the file. This can be achieved by using the **asdcplib-test** software utility, and subtracting one from the ContainerDuration value, as shown below.

```

$ asdcplib-test -i -v PerfectMovie-j2c-pt.mxf
File essence type is JPEG 2000 pictures.
  ProductUUID: 43059a1d-0432-4101-b83f-736815acf31d
  ProductVersion: Unreleased 1.1.13
  CompanyName: DCI
  ProductName: asdcplib
  EncryptedEssence: No
  AssetUUID: 0e676fb1-951b-45c4-8334-ed2c59199815
  Label Set Type: SMPTE
  AspectRatio: 2048/1080
  EditRate: 24/1
  StoredWidth: 2048
  StoredHeight: 1080
    Rsize: 3
    Xsize: 2048
    Ysize: 1080
  XOffset: 0
  YOffset: 0
  XTsize: 2048
  YTsize: 1080
  XTOffset: 0
  YTOffset: 0
  ContainerDuration: 240
Color Components:
  11.1.1
  11.1.1
  11.1.1
Default Coding (16): 01040001010503030000778888888888
Quantization Default (33): 227f187f007f007ebc76ea76ea76bc6f4c6f4c6f645803580358455fd25fd25f61

```

2. Using the **asdcplib-test** software utility, extract the first and the last frames of content from the Track file.

```

$ asdcplib-test -x first -d 1 -f 0 PerfectMovie-j2c-pt.mxf
$ asdcplib-test -x last -d 1 -f 239 PerfectMovie-j2c-pt.mxf
$ ls
first000000.j2c
last000239.j2c
PerfectMovie-j2c-pt.mxf

```

3. Verify that the first and the last frames of content decode completely, and without errors. Failure to correctly decode either frame is cause to fail this test. This can be achieved by using JPEG 2000 decoding software. An example is shown below. (Note that the output of the **j2k-scan** program is long and has been truncated here for brevity. Please see Section C.5 for a detailed example.)

```
$ j2k-scan frame000000.j2c
digital cinema profile: none
rsiz capabilities: standard
pixel offset from top-left corner: (0, 0)
tile width/height in pixels: (2048, 1080)
image width/height in tiles: (1, 1)
tile #1
  coding style: 1
  progression order: Component-Position-Resolution-Layer
  POC marker flag: 0
  number of quality layers: 1
  rate for layer #1: 0.0
  multi-component transform flag: 1
...
```

## Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.3.3.2
[SMPTE-336M]	
[SMPTE-422M-2006]	

Test Equipment
asdcv-test
j2c-scan

## 4.4.6. Audio Track File Frame Boundary

### Objective

The Audio Track File is required to begin and end with complete frames that are associated with its Image Track File to allow for a clean transition between reels. The audio data within the Track File shall be wrapped using KLV on an image frame boundary.

### Procedures

Verify that exactly the expected number of Audio bytes are embedded within each KLV encoded triplet for each frame of the Audio Track File. This can be achieved by using the software command **klvwalk** to display the length of every WAVEssence set (UL value 060e2b34.0102.0101.0d010301.16010101) and checking that each frame contains the appropriate number of bytes. The expected number of Audio Bytes per frame can be calculated by using the formula  $len = BPS * Ch * SPF$ , where BPS is the number of Bytes Per Sample (BPS=3), Ch is the number of Audio Channels in the DCP, and SPF is the number of Samples Per Frame value taken from Table 4.2.

If any frame has an actual len that differs from the expected value, calculated from the formula, this is cause to fail this test.

The example below shows eight frames of a composition containing six channels of 48kHz samples at 24fps, completely wrapped in KLV triplets ( $3 * 6 * 2000 = 36000$ ).

```
$klvwalk PerfectMovie-pcm-pt.mxf
...
060e2b34.0102.0101.0d010301.16010101 len: 36000 (WAVEssence)
060e2b34.0102.0101.0d010301.16010101 len: 36000 (WAVEssence)
060e2b34.0102.0101.0d010301.16010101 len: 36000 (WAVEssence)
060e2b34.0102.0101.0d010301.16010101 len: 36000 (WAVEssence)
060e2b34.0102.0101.0d010301.16010101 len: 36000 (WAVEssence)
060e2b34.0102.0101.0d010301.16010101 len: 36000 (WAVEssence)
060e2b34.0102.0101.0d010301.16010101 len: 36000 (WAVEssence)
060e2b34.0102.0101.0d010301.16010101 len: 36000 (WAVEssence)
...
```

The possible values for the Samples/Frame are shown in table below.

**Table 4.2. Audio Samples Per Frame**

FPS	Sample Rate	Samples/Frame
24	48 kHz	2000
24	96 kHz	4000
48	48 kHz	1000
48	96 kHz	2000

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.3.4.2

<b>Test Equipment</b>
klvwalk

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Discontinued May 8 2009

## 4.5. Essence

### 4.5.1. Image Structure Container and Image Container Format

#### Objective

- Verify that the images contained in the Track Files conform to an Image Structure Container that consists of either 4K (4096x2160) (Operational Level 1) or 2K (2048x1080) (Operational Level 2 and 3). It is expected that the image structure shall use one of the two containers such that either the horizontal or vertical resolution is filled.
- Verify that both the horizontal and vertical dimensions of the image structure container are divisible by four for Level 1, or two for Level 2 and 3 image structures. This ensures that the image can be centered correctly.
- Verify that the bit depth for each code value for a color component shall be 12 bits. This yields 36 bits per pixel.

#### Procedures

1. Using the software command **klvwalk**, locate the RGBAEssenceDescriptor set and record the StoredWidth, StoredHeight, and AspectRatio values within. The failure to meet any of the following conditions is cause to fail this test:
  - a. Verify that the first number (numerator) of the AspectRatio field is the same as the StoredWidth value.
  - b. Verify that the second number (denominator) of the AspectRatio field is the same as the StoredHeight value.
  - c. Verify that exactly one of the StoredWidth or StoredHeight values are equal to the Maximum Horizontal Pixels or Maximum Vertical Pixels values from Table 4.3: Image Structure Operational Levels below.
  - d. Verify that both the StoredWidth and StoredHeight values are equal to, or less than, the Maximum Horizontal Pixels or Maximum Vertical Pixels values, respectively, from Table 4.3 below.
  - e. Verify that both the StoredWidth and StoredHeight values are exactly divisible by two for a 2K file, and four for a 4K file.

An example of the RGBAEssenceDescriptor set is shown below:

```
$ klvwalk -r PerfectMovie-j2c-pt.mxf
...
060e2b34.0253.0101.0d010101.01012900 len: 169 (RGBAEssenceDescriptor)
  InstanceUID = 82141918-celb-47a5-ac13-c47cfeb2e51a7
  GenerationUID = 00000000-0000-0000-0000-000000000000
  Locators:
  SubDescriptors:
    92e96e5e-6bef-4985-8117-7dfa541f96fa
      LinkedTrackID = 2
      SampleRate = 24/1
      ContainerDuration = 240
      EssenceContainer = 060e2b34.0401.0107.0d010301.020c0100
        Codec = 060e2b34.0401.0109.04010202.03010103
        FrameLayout = 0
        StoredWidth = 2048
        StoredHeight = 1080
```



```

        AspectRatio = 2048/1080
        ComponentMaxRef = 4095
        ComponentMinRef = 0
...

```

The valid Image Structure Container values are shown in table below.

**Table 4.3. Image Structure Operational Levels**

Operational Level	Maximum Horizontal Pixels	Maximum Vertical Pixels	Frames per Second
1	4096	2160	24
2	2048	1080	48
3	2048	1080	24

- Using the software commands **asdcp-test** and **j2k-scan**, extract an image frame from the file and verify that the bit depth for each component is 12 bits. A component bit-depth value other than 12 shall be cause to fail this test.

An example of this operation is shown below:

```

$ asdcp-test -d 1 -x frame j2c/PerfectMovie-j2c-pt.mxf
$ j2c-scan frame_000001.j2c
coding parameters
  digital cinema profile: none
  rsiz capabilities: standard
  pixel offset from top-left corner: (0, 0)
  tile width/height in pixels: (2048, 1080)
  image width/height in tiles: (1, 1)
...

```

## Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	3.2.1.2, 3.2.1.3, 3.2.1.7
[SMPTE-428-1]	

Test Equipment
klvwalk
asdcp-test
j2c-scan

## 4.5.2. Image Compression Standard & Encoding Parameters

### Objective

Verify that the image encoding parameters in a Picture Track File conform to [SMPTE-429-4].

### Procedures

1. Verify that the UL value in the Codec field of the MXF RGBAEssenceDescriptor (see 4 in Example 4.8) is one of:  
060e2b34.0401.0109.04010202.03010103 (2K images) or  
060e2b34.0401.0109.04010202.03010104 (4K images).

If the UL value does not match one of those listed above, or is the wrong value for the contained essence, this is cause to fail the test.

2. Using a software command such as **asdcp-test**, extract all the frames in the Track File to a directory. An example is shown below.

```
$ asdcp-test -x frame j2c/PerfectMovie-j2c-pt.mxf
$ ls j2c
frame000000.j2c      frame000057.j2c      frame000124.j2c      frame000191.j2c
frame000001.j2c      frame000058.j2c      frame000125.j2c      frame000192.j2c
frame000002.j2c      frame000059.j2c      frame000126.j2c      frame000192.j2c
frame000003.j2c      frame000060.j2c      frame000127.j2c      frame000194.j2c
...
```

3. Verify that every frame is correctly JPEG 2000 encoded as described in [ISO 15444-1]. Verify that the proper JPEG 2000 encoding parameters as specified in [ISO PDAM-1] were used. The Codestream Specifications for 2K and 4K distributions are listed in [DCI-DCSS-1-1], section 4.4. This can be achieved by using JPEG 2000 decoding software. An example is shown below. (Note that the output of the **j2k-scan** program is long and has been truncated here for brevity. Please see Section C.5 for a detailed example.) If any frame fails to correctly decode or does not conform to the appropriate Codestream Specifications, this is cause to fail the test.

```
$ j2k-scan frame000000.j2c
digital cinema profile: none
rsiz capabilities: standard
pixel offset from top-left corner: (0, 0)
tile width/height in pixels: (2048, 1080)
image width/height in tiles: (1, 1)
tile #1
  coding style: 1
  progression order: Component-Position-Resolution-Layer
  POC marker flag: 0
  number of quality layers: 1
    rate for layer #1: 0.0
  multi-component transform flag: 1
...
```

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1  
Used May 8 2009

## Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [ISO-15444-1] [ISO-15444-1-AMD-1]	3.2.1.5, 4.2, 4.4

Test Equipment
asdcp-test OpenJPEG

### 4.5.3. Audio Characteristics

#### Objective

Sound Track Files shall conform to the specifications given in [SMPTE-428-2] and [SMPTE-428-3], and be constrained as specified in [SMPTE-429-2]. A Sound Track File shall contain linear PCM audio sampled at 48000 or 96000 samples per second, 24 bits per sample. The file shall contain no more than 16 channels of audio.

#### Procedures

Using the software command **klvwalk**, locate the WaveAudioDescriptor set which starts with the Universal Label (UL) of 060e2b34.0253.0101.0d010101.01014800. An example is shown below.

```
$ klvwalk -r PerfectMovie-pcm-pt.mxf
...
060e2b34.0253.0101.0d010101.01014800 len:      134 (WaveAudioDescriptor)
InstanceUID = e1c4c755-2c3e-4274-a3bf-581aadd63a4b
GenerationUID = 00000000-0000-0000-0000-000000000000
Locators:
SubDescriptors:
LinkedTrackID = 2
SampleRate = 24/1
ContainerDuration = 480
EssenceContainer = 060e2b34.0401.0101.0d010301.02060100
Codec = 00000000.0000.0000.00000000.00000000
AudioSamplingRate = 48000/1
Locked = 0
AudioRefLevel = 0
ChannelCount = 6
QuantizationBits = 24
DialNorm = 0
BlockAlign = 18
SequenceOffset = 0
AvgBps = 144000
...
```

Verify the following:

1. The EssenceContainer field has a value of 060e2b34.0401.0101.0d010301.02060100. Any other value is cause to fail this test.
2. The AudioSamplingRate field has a value of either 48000/1 or 96000/1. Any deviation from these values is cause to fail this test.
3. The ChannelCount field has a value of no fewer than 1 and no greater than 16. Any deviation from these values is cause to fail this test.
4. The QuantizationBits field has a value of 24. Any other value is cause to fail this test.
5. The BlockAlign field is exactly the value of ChannelCount\*3. Any other value is cause to fail this test.
6. The AvgBps field is exactly the value of the AudioSamplingRate\*3. Any other value is cause to fail this test.

**Supporting Materials**

<b>Reference Document ID</b>	<b>Reference Document Section(s)</b>
[DCI-DCSS-1-1]	3.3.2.2, 3.3.4.1
[SMPTE-428-2]	
[SMPTE-428-3]	
[SMPTE-429-2]	

<b>Test Equipment</b>
klvwalk

## 4.5.4. Timed Text Resource Encoding

### Objective

Timed text essence must conform to the following standards:

1. Timed text descriptions in XML must conform to [SMPTE-428-7].
2. Font resources must conform to [ISO-144496].
3. Sub-picture resources must conform to [ISO-15948].

### Procedures

1. Extract the Timed Text Resource and any Ancillary Resources from the Track File.
2. Verify that the Timed Text Resource is an XML document that can be validated using the schema from [SMPTE-428-7]. If the XML validation produces errors, this is cause to fail this test.

```
$ schema-check testfile.xml S428-7-2007.xsd
$
```

3. Verify that any font resources are valid according to [ISO-144496]. If the font validation produces errors, this is cause to fail this test.

```
$ ftlint 1 font_file.otf
font_file.otf: OK.
$
```

4. Verify that any subpicture resources are valid according to [ISO-15948]. The subpicture must be of PNG format, decode without errors, and the size (geometry) must be smaller than, or equal to, that of the main picture. If the png file causes **identify** to report errors, or if the geometry of the PNG is greater than that of the main picture, this is cause to fail this test.

```
$ identify -verbose subpicture_0001.png
Image: subpicture_0001.png
Format: PNG (Portable Network Graphics)
Geometry: 120x420
Class: DirectClass
Colorspace: RGB
Type: GrayscaleMatte
Depth: 8 bits
...
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	3.4.2.2, 4.4.3.2, 3.4.3.4

Version  
NEWER  
Version  
0 ARCHIVE ONLY  
VERSION AVAILABLE  
Used May 8 2009

Reference Document ID	Reference Document Section(s)
[ISO-144496] [ISO-15948] [SMPTE-428-7]	

Test Equipment
schema-check ftlint identify

## 4.6. Digital Cinema Package

### 4.6.1. DCP Integrity

#### Objective

- Verify that the Volume Asset Map is present, correctly formatted, and correctly located in the filesystem.
- Verify that for all the Packing Lists found in the Asset Map file, all of the assets referenced in each Packing List are present and are valid (i.e., each Referenced Asset's file size and Message Digest are correct).

File Integrity will be guaranteed by applying the SHA-1 hashing algorithm [RFC3174] to each asset included in the DCP. The resulting message digest is Base64 encoded and included in the Packing List file.

- Verify that for all the Composition Playlists found in each Packing List, the Referenced Assets exist in the Packing List file.

#### Procedures

1. Validate the Format of the Volume Asset Map file by executing the test procedure Section 4.1.1: Asset Map File.
2. Validate the Format of the Volume Index file by executing the test procedure Section 4.1.2: Volume Index File.
3. Validate the Format of each Packing List file by executing the test procedure Section 4.2.1: Packing List File.
4. Validate the Signature of each Packing List file by executing the test procedure Section 4.2.2: Packing List Signature Validation.
5. For each Packing List file (e.g. PerfectMovie.pkl.xml) in the Asset Map:
  - a. Open the Packing List and for each Asset Id contained within:
    - i. Locate the Referenced Asset in the filesystem and compare its file size with the value listed in the <Size> element of the <Asset> element. Inconsistency is cause to fail this test.
    - ii. Calculate the Message Digest of the Referenced Asset and encode the result in Base64. Compare the result with the value listed in the <Hash> element of the <Asset> element. Inconsistency is cause to fail this test. The following is an example using the **asdcptest** software utility:

```
$ asdcptest -t PerfectMovie-j2c-pt.mxf  
t0MirEHOVFF4Mi1IP0iYVjrvb14= PerfectMovie-j2c-pt.mxf
```

6. Validate the Format of each Composition Playlist file by executing the test procedure Section 4.3.1: Composition Playlist File.
7. Validate the Signature of each Composition Playlist file by executing the test procedure Section 4.3.2: Composition Playlist Signature Validation.
8. For each Composition Playlist (e.g. PerfectMovie.cpl.xml) in the Asset Map:



a. Open the Composition Playlist and for each Asset Id contained within:

i. Locate the Asset Id in the Packing List file. Any missing Asset Ids are cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.2.2.6, 5.3.1.9, 5.5.2.3, 5.5.3.2, 9.7.5
[PKCS-1]	
[RFC-3174]	
[SMPTE-429-8-2007]	

Test Equipment
asdcp-test

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

**Page Intentionally Left Blank**

---

# Chapter 5. Common Security Features

This chapter contains test procedures of security features that apply to more than one type of device. Procedures are given for Type 1 and Type 2 Secure Processing Block (SPB) physical security requirements, Intra-theater communications and security log reporting.

Version 1.0 ARCHIVED ONLY  
NEWER VERSION AVAILABLE  
Version 1.0 released May 8 2009

## 5.1. SPB Security Features

The test procedures in this section apply to any device or component that is classified as a Type 1 or Type 2 SPB.

### 5.1.1. SPB Digital Certificate

#### Objective

Verify that each SPB carries exactly one d-cinema certificate [SMPTE 430-2], and that each SPB certificate correctly designates via role identifiers, in the certificate Common Name field, the specific SEs that are contained within the SPB. Verify that the certificate Common Name carries information that identifies the physical device (e.g., a serial number).

#### Procedures

1. Obtain the device's certificate from the manufacturer (or directly by capturing the TLS "Server Hello" message).
2. Using manufacturer-supplied documentation, compile the list of role identifiers (per [SMPTE 430-2-2006]) corresponding to the set of Security Entities (SE) that exist in the device.
3. Compare the list from step 2 above to the role set in the Common Name field of the certificate obtained in step 1 above. Mismatched lists are cause to fail the test.
4. Verify that the certificate Common Name field contains some value that is clearly indicated on the physical device (e.g., a serial number), or that the manufacturer can use to identify the physical device. Failure to meet this condition is cause to fail the test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.1

## 5.1.2. SPB Type 2 Security Perimeter

### Objective

Verify that a Type 2 SPB module implements hardware module perimeter protection that prevents access to internal circuitry and detects and records opening of the module perimeter.

### Procedures

Using manufacturer-supplied documentation:

1. Define the physical area of the SPB in the system.
2. Verify that the SPB contains all sensitive circuits (i.e., all plaintext signals and plaintext interfaces).
3. Verify that entry into the SPB perimeter is logged to a persistent log in secure silicon (see also Section 5.1.3).
4. Failure to verify condition 2 or 3 above is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.4

### 5.1.3. SPB Type 2 Secure Silicon

#### Objective

Verify that secure silicon level protection is used within a Type 2 SPB, with such protection continuously provided (including if powered down) for all SPB Critical Security Parameters (CSP). Verify that the device meets FIPS 140-2 Level 3, area five (*physical security*).

#### Procedures

1. Using manufacturer-supplied documentation, verify that the facilities for perimeter protection and marriage to Type 1 SPB are implemented using a self-contained, always-on processing module.
2. Verify that the processing module is entirely contained within the SPB perimeter.
3. Verify that the processing module meets FIPS 140-2 Level 3, area five (*physical security*).
4. Failure to verify all conditions above is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.2

## 5.2. Intra-Theater Communication

The procedures in this section apply to devices which can initiate or respond to TLS session requests using TCP port 1173.

### 5.2.1. TLS Session Initiation

#### Objective

Verify that once started, the Security Manager (SM) establishes a TLS session with all SPB devices it is configured to recognize. Verify that each TLS session is persistent.

#### Procedures

If the Test Subject is a Security Manager device:

1. Using manufacturer-supplied documentation, verify that the system is designed to maintain a persistent TLS connection with configured SPBs.
2. Configure the Test Subject to recognize one or more SPB devices, such as a projector with a Link Decryptor Block.
3. Construct a test network consisting of an Ethernet switch, the Test Subject, SPB(s) and a network analyzer.
4. With only the Ethernet switch and network analyzer on, turn on each SPB device and allow the device to come to an idle state.
5. Turn on the Test Subject. Verify that for each SPB in the system, the SM establishes a TLS session soon after startup.
6. Verify that each TLS session is maintained over time. A minimum monitoring period of one hour of system idle time should be observed.
7. Disconnect an SPB from the test network long enough to cause the SM to close the connection (use manufacturer-supplied documentation to determine the appropriate delay).
8. Reconnect the disconnected SPB. Verify that the SM re-establishes a TLS session within the time documented by the manufacturer.
9. Failure to verify any of the conditions listed above is cause to fail this test.

If the Test Subject is a Type 1 SPB:

1. Using manufacturer-supplied documentation, verify that the system is designed to accept TLS connections on TCP port 1173.
2. Using manufacturer-supplied documentation, verify that the system is designed not to initiate TLS connections to other systems on TCP port 1173.
3. Construct a test network consisting of an Ethernet switch, the Test Subject and a network analyzer.
4. With only the Ethernet switch and network analyzer on, turn on the Test Subject and allow the device to come to an idle state. A connection request from the Test Subject using TCP port 1173 is cause to fail this test.

Version 1.1 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

5. Verify that the Test Subject accepts a connection request on TCP port 1173 from an SM or ASM requester simulator.

If the Test Subject is a Type 2 SPB:

1. Construct a test network consisting of an Ethernet switch, the Test Subject and a network analyzer.
2. With only the Ethernet switch and network analyzer on, turn on the Test Subject and allow the device to come to an idle state. A connection request from the Test Subject using TCP port 1173 is cause to fail this test.
3. Verify that the Test Subject accepts a connection request on a TCP port designated by the manufacturer from an SM or ASM simulator.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3
[SMPTE-430-6]	

Test Equipment
Network Analyzer
Ethernet Switch
asm-requester

## 5.2.2. Auditorium Security Messages

Auditorium Security Messages (ASM) are used to communicate runtime security information between a Security Manager (SM) and a remote Link Decryptor Block (LDB). The following test procedures apply to any device which can initiate (TLS client) or terminate (TLS server) a TLS session.

To test a device which implements ASM, it will be necessary to use an ASM simulator program or any suitably instrumented peer device. To simplify the descriptions in the procedures below, the language assumes the use of an ASM simulator. A detailed description of a reference ASM simulator is given in Appendix D: *ASM Simulator*.

### 5.2.2.1. Auditorium Security Message Support

#### Objective

Verify that Auditorium Security Messages (ASM) are implemented per [SMPTE 430-6]. Verify correct operation of applicable messages. Verify use of the IANA-assigned TCP port number 1173.

#### Procedures

1. Start a TLS session between the Test Subject and an ASM simulator.
2. Verify that the responder accepted the TCP open request on TCP port 1173.
3. Using manufacturer-supplied documentation, verify that TCP port 1173 is not used for any protocol other than ASM.
4. Verify that certificates were exchanged by both sides and that the certificates are valid per [SMPTE 430-2-2006].
5. Verify that a QuerySPB message occurs within 1 minute of TLS startup (as reported by the ASM simulator).

For testing responders, the **asm-requester** command must send a QuerySPB message:

```
$ asm-requester (... standard options ...) \  
--messagetype QuerySPB
```

For testing requesters, the **asm-responder** command must respond to the QuerySPB message:

```
$ asm-requester (... standard options ...) \  
--messagetype QuerySPB
```

6. Failure to verify all conditions above is cause to fail the test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3



Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Released May 8 2009

Reference Document ID	Reference Document Section(s)
[SMPTE-430-6]	

Test Equipment
asm-requester
asm-responder

### 5.2.2.2. ASM Failure Behavior

#### Objective

1. Verify that an ASM responder provides a response within the recommended delay interval.
2. Verify that an ASM responder provides appropriate security alert response codes for significant security events.
3. Verify that an ASM requester continues to operate normally when it receives a ResponderBusy response.

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM responder.
2. Command the simulator to return all ASM requests with ResponderBusy. Verify that the Test Subject attempts to function normally.

```
$ asm-responder (... standard options ...) \  
--respond-with "Busy"
```

3. Failure to verify the conditions above is cause to fail the test.
4. Command the simulator to hold responses until TLS renegotiation. Note whether the Test Subject restarts the TLS session.

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM requester and the Test Subject.
2. Command the simulator to issue an arbitrary sequence of requests.

```
$ asm-requester (... standard options ...) \  
--messagetype <message-type>
```

Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].

3. For Test Subjects which have field-operable perimeter access, open an access panel and verify that the Subject then responds to all ASM requests with error code "2".
4. For Test Subjects which can participate in marriage to a companion device and be accessed via ASM in the divorced state, place the device in the "divorced" state and verify that the Subject then responds to all ASM requests with error code "2".
5. Failure to verify the conditions above is cause to fail the test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.3.2

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Released May 8 2009

Reference Document ID	Reference Document Section(s)
[SMPTE-430-6]	

Test Equipment
asm-requester
asm-responder

### 5.2.2.3. ASM "RRP Invalid"

#### Objective

Verify that an ASM "RRP Invalid" response is supported.

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM responder simulator.
2. Command the ASM responder simulator to return all ASM requests with RRP Invalid. Verify that the Test Subject attempts to function normally.
3. Failure to verify the conditions above is cause to fail the test.

```
$ asm-responder (... standard options ...) --respond-with Invalid
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5
[SMPTE-430-6]	

Test Equipment
asm-responder

### 5.2.2.4. ASM "GetTime"

#### Objective

Verify that the Test Subject implements the GetTime command per [SMPTE 430-6].

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM responder simulator.
2. Using manufacturer-supplied documentation, command the Test Subject to issue a GetTime request. Observe that the request is accepted by the simulator without error.
3. Failure of the device to implement the GetTime command is cause to fail the test.

```
$ asm-responder (... standard options ...)
```

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM requester simulator and the Test Subject.
2. Command the simulator to issue a GetTime request. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
3. Failure to verify the conditions above is cause to fail the test.
4. Record the difference measured between the time value returned and real time as reported by the reference clock.

```
$ asm-requester (... standard options ...) \  

  --messagetype GetTime
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-6]	9.4.5

Test Equipment
asm-responder asm-requester Accurate Real-Time Clock

### 5.2.2.5. ASM "GetEventList"

#### Objective

Verify that the Test Subject implements the GetEventList command per [SMPTE 430-6].

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM responder simulator.
2. Using manufacturer-supplied documentation, command the Test Subject to issue a GetEventList request. Observe that the request is accepted by the simulator without error.
3. Failure of the device to implement the GetEventList command is cause to fail the test.

```
$ asm-responder (... standard options ...) \  

  --preload-log-event SPBOpen.xml \  

  --preload-log-event SPBClose.xml \  

  --preload-log-event SPBMarriage.xml \  

  --preload-log-event SPBDivorce.xml \  

  --preload-log-event SPBQuery.xml
```

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM requester simulator and the Test Subject.
2. Command the simulator to issue a GetEventList request. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
3. Failure to verify the conditions above is cause to fail the test.

```
$ asm-requester (... standard options ...) \  

  --messagetype GetEventList \  

  --start-time 2007-07-23T19:00:00-00:00 \  

  --end-time 2007-07-23T121:00:00-00:00
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-6]	9.4.5

Test Equipment
asm-requester asm-responder

## 5.2.2.6. ASM "GetEventID"

### Objective

Verify that the Test Subject implements the GetEventID command per [SMPTE 430-6].

### Procedures

Each GetEventID procedure call returns an XML document with a top-level element LogRecordElement. See Example 5.2 for more information about this data type.

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM responder simulator.
2. Using manufacturer-supplied documentation, command the Test Subject to issue a GetEventID request. Observe that the request is accepted by the simulator without error.
3. Failure of the device to implement the GetEventID command is cause to fail the test.

```
$ asm-responder (... standard options ...) \
  --preload-log-event SPBOpen.xml \
  --preload-log-event SPBClose.xml \
  --preload-log-event SPBMarriage.xml \
  --preload-log-event SPBDivorce.xml \
  --preload-log-event SPBQuery.xml
```

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM requester simulator and the Test Subject.
2. Command the simulator to issue a GetEventID request. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
3. Verify that the text value returned is a valid XML document per [SMPTE 430-4].
4. Failure to verify the conditions above is cause to fail the test.

```
$ asm-requester (... standard options ...) \
  --messagetype GetEventID \
  --message-id <event-id>
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5
[SMPTE-430-4]	
[SMPTE-430-6]	

<b>Test Equipment</b>
asm-responder
asm-requester

Version  
NEWER  
Versi  
0 ARCHIVE ONLY  
SION AVAILABLE  
sed May 8 2009



### 5.2.2.7. ASM "LEKeyLoad"

#### Objective

Verify that the Test Subject implements the LEKeyLoad command per [SMPTE 430-6].

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM responder simulator.
2. Using manufacturer-supplied documentation, command the Test Subject to issue a LEKeyLoad request. Observe that the request is accepted by the simulator without error.
3. Repeat step 2 after commanding the simulator to refuse LEKeyLoad requests. Verify that the Test Subject attempts to operate normally.
4. Failure of the device to implement the LEKeyLoad command is cause to fail the test.

```
$ asm-responder (... standard options ...)
```

```
$ asm-responder (... standard options ...) \  
--respond-with Busy
```

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM simulator and the Test Subject.
2. Command the simulator to issue a LEKeyLoad request. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
3. Command the simulator to flood the Test Subject with LEKeyLoad messages. Verify that the Subject responds to overflow with an appropriate error.
4. Failure to verify the conditions above is cause to fail the test.

```
$ asm-requester (... standard options ...) \  
--messagetype LEKeyLoad
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5
[SMPTE-430-6]	

<b>Test Equipment</b>
asm-requester

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Discontinued May 8 2009

### 5.2.2.8. ASM "LEKeyQueryID"

#### Objective

Verify that the Test Subject implements the LEKeyQueryID command per [SMPTE 430-6].

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM simulator.
2. Using manufacturer-supplied documentation, command the Test Subject to issue a LEKeyQueryID request. Observe that the request is accepted by the simulator without error.
3. Failure of the device to implement the LEKeyQueryID command is cause to fail the test.

```
$ asm-responder (... standard options ...)
```

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM simulator and the Test Subject.
2. Command the simulator to issue a LEKeyQueryID request for a known present key ID. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
3. Command the simulator to issue a LEKeyQueryID request for a known bogus key ID. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
4. Failure to verify the conditions above is cause to fail the test.

```
$ asm-requester (... standard options ...) \  

  --messagetype LEKeyQueryID \  

  --messagetype-id <key-id>
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-6]	9.4.5

Test Equipment
asm-requester

### 5.2.2.9. ASM "LEKeyQueryAll"

#### Objective

Verify that the Test Subject implements the LEKeyQueryAll command per [SMPTE 430-6].

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM simulator.
2. Using manufacturer-supplied documentation, command the Test Subject to issue a LEKeyQueryAll request. Observe that the request is accepted by the simulator without error.
3. Failure of the device to implement the LEKeyQueryAll command is cause to fail the test.

```
$ asm-responder (... standard options ...)
```

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM simulator and the Test Subject.
2. Command the simulator to issue a LEKeyQueryAll request. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
3. Failure to verify the conditions above is cause to fail the test.

```
$ asm-requester (... standard options ...) \  

  --messagetype LEKeyQueryAll
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5
[SMPTE-430-6]	

Test Equipment
asm-requester
asm-responder

### 5.2.2.10. ASM "LEKeyPurgeID"

#### Objective

Verify that the Test Subject implements the LEKeyPurgeID command per [SMPTE 430-6].

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM simulator.
2. Using manufacturer-supplied documentation, command the Test Subject to issue a LEKeyPurgeID request. Observe that the request is accepted by the simulator without error.
3. Failure of the device to implement the LEKeyPurgeID command is cause to fail the test.

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM simulator and the Test Subject.
2. Command the simulator to issue a LEKeyPurgeID request. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
3. Failure to verify the conditions above is cause to fail the test.

```
$ asm-requester (... standard options ...) \  

  --messagetype LEKeyPurgeID \  

  --messagetype-id <key-id>
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5
[SMPTE-430-6]	

Test Equipment
asm-requester

### 5.2.2.11. ASM "LEKeyPurgeAll"

#### Objective

Verify that the Test Subject implements the LEKeyPurgeAll command per [SMPTE 430-6].

#### Procedures

If the Test Subject is an ASM requester:

1. Initiate an ASM session between the Test Subject and the ASM simulator.
2. Using manufacturer-supplied documentation, command the Test Subject to issue a LEKeyPurgeAll request. Observe that the request is accepted by the simulator without error.
3. Failure of the device to implement the LEKeyPurgeAll command is cause to fail the test.

```
$ asm-responder (... standard options ...)
```

If the Test Subject is an ASM responder:

1. Initiate an ASM session between the ASM simulator and the Test Subject.
2. Command the simulator to issue a LEKeyPurgeAll request. Verify that the Test Subject responds within the 2 second maximum delay period recommended by [SMPTE 430-6].
3. Failure to verify the conditions above is cause to fail the test.

```
$ asm-requester (... standard options ...) \  

  --messagetype LEKeyPurgeALL
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5
[SMPTE-430-6]	

Test Equipment
asm-requester
asm-responder

## 5.3. Event Logs

Secure Processing Block (SPB) modules are required to provide event log reports on demand. The log reports are XML documents (see Section 3.1) having a structure defined by [SMPTE 430-4]. This section will describe the report format and present procedures for testing general operational requirements for event logging.

### Note

The method of generating a log report will vary between implementations. Consult the manufacturer's documentation for log generation instructions.

### 5.3.1. Log Report Format

Standard d-cinema log reports are encoded as XML documents per [SMPTE-430-4]. The reports consist of a preamble which, identifies the device that created the report, and a sequence of log records. In log reports which contain security events (Security Event Logs), some of the log records may contain XML Signature elements. The report format includes many unique security features; the reader should study [SMPTE-430-4] in detail to understand how logs authentication works.

The following subsections detail the major features of a log report.

#### 5.3.1.1. Log Report Preamble

The log report begins with `reportDate` and `reportingDevice` elements. The contents of the elements identify the time the log was created and the device that created the log.

#### Example 5.1. Log Report Preamble Example

```
<?xml version="1.0" encoding="UTF-8"?>
<LogReport ❶
  xmlns="http://www.smpte-ra.org/schemas/430-4/2007/LogRecord" ❷
  xmlns:dcml="http://www.smpte-ra.org/schemas/433/2006/dcmlTypes" >
  <reportDate>2007-05-04T09:30:47-08:00</reportDate> ❸

  <reportingDevice> ❹
    <dcml:DeviceIdentifier idtype="CertThumbprint">YmVsc3dpY2tAZW50ZXJ0ZWNoLmNvbQ==
    </dcml:DeviceIdentifier>
    <dcml:DeviceTypeID
      scope="http://www.smpte-ra.org/schemas/433/2006/DCMLTypes/#DeviceTypeTokens"
    >SM</dcml:DeviceTypeID>
    <dcml:DeviceSerial>000000042</dcml:DeviceSerial> ❺
    <dcml:ManufacturerCertID>rlpve6MSncWouNIpFcTSIhk6w2A=</dcml:ManufacturerCertID>
    <dcml:DeviceCertID>9czqa+0orIADHDIYxAkn/IcmZ3o=</dcml:DeviceCertID> ❻
    <dcml:ManufacturerName>Acme Digital Cinema Inc.</dcml:ManufacturerName>
    <dcml:DeviceName>Mojo Media Block</dcml:DeviceName>
    <dcml:ModelNumber>MB-3000</dcml:ModelNumber>
    <dcml:VersionInfo>
      <dcml:Name>Bootloader</dcml:Name>
      <dcml:Value>1.0.0.0</dcml:Value>
      <dcml:Name>Security Module</dcml:Name>
      <dcml:Value>3.4.2.1</dcml:Value>
    </dcml:VersionInfo>
  </reportingDevice>
```

## Log Record Preamble Description

- 1 The LogReport element is the root element of a log report document.
- 2 The Log Report and DCML namespaces are used.
- 3 This value gives the date on which this report document was generated.
- 4 This structure identifies the device that generated this report.
- 5 The serial number of generating device.
- 6 The certificate thumbprint (per [SMPTE 430-2]) of the generating device.

### 5.3.1.2. Log Report Record

Each event contained in the log report is encoded as a LogRecordElement element. This element type has three major sub-elements: LogRecordHeader, LogRecordBody and LogRecordSignature. The first two are shown in the example below, the last is the subject of the next section.

Elements of this type are also used directly (without a containing LogReport parent element) as the return value from an ASM GetEventID procedure (see Section 5.2.2.6.) Because ASM procedures are executed exclusively via TLS with a trusted peer, the LogRecordSignature element is not needed.

#### Example 5.2. Log Report Record Example

```

<LogRecordElement 1
  xmlns="http://www.smpte-ra.org/schemas/430-4/2007/LogRecord"
  xmlns:dcml="http://www.smpte-ra.org/schemas/433/2006/dcmlTypes">
  <LogRecordHeader>
    <EventID>urn:uuid:8a221dfc-f5c6-426d-a2b8-9f6ff1cc6e31</EventID> 2
    <TimeStamp>2005-12-17T10:45:00-05:00</TimeStamp> 3
    <EventSequence>1000003</EventSequence> 4
    <DeviceSourceID>
      <dcml:PrimaryID idtype="CertThumbprint">VGVzdCBDZXJ0aWZpY2F0ZSBuHVtYnByaW50
      </dcml:PrimaryID>
    </DeviceSourceID>
    <EventClass>http://www.smpte-ra.org/430.5/2007/SecurityLog</EventClass> 5
    <EventType
      scope="http://www.smpte-ra.org/430.5/2007/SecurityLog/#EventTypes">Key</EventType> 6
    <contentId>urn:uuid:733365c3-2d44-4f93-accd-43cb39b0cedf</contentId> 7
    <previousHeaderHash>9czqa+0orIADHDIYxAkn/IcmZ3o=</previousHeaderHash> 8
    <recordBodyHash>9czqa+0orIADHDIYxAkn/IcmZ3o=</recordBodyHash> 9
  </LogRecordHeader>
  <LogRecordBody>
    <EventID>urn:uuid:8a221dfc-f5c6-426d-a2b8-9f6ff1cc6e31</EventID>
    <EventSubType
      scope="http://www.smpte-ra.org/430.5/2007/SecurityLog/#EventSubTypes-key">
      KDMKeysReceived
    </EventSubType> 10
    <Parameters> 11
      <dcml:Parameter>
        <dcml:Name>SignerID</dcml:Name>
        <dcml:Value xsi:type="ds:DigestValueType">rlpve6MSncWouNIpFcTShk6w2A</dcml:Value>
      </dcml:Parameter>
    </Parameters>
    <Exceptions> 12
      <dcml:Parameter>
        <dcml:Name>KDMFormatError</dcml:Name>
        <dcml:Value xsi:type="xs:string">XML validation failed on line 36</dcml:Value>
      </dcml:Parameter>
    </Exceptions>
    <ReferencedIDs> 13
      <ReferencedID>
        <IDName>CompositionID</IDName>
        <IDValue>urn:uuid:64bb6972-13a0-1348-a5e3-ae45420ea57d</IDValue>

```



```

    </ReferencedID>
    <ReferencedID>
      <IDName>KeyDeliveryMessageID</IDName>
      <IDValue>urn:uuid:64bb6972-13a0-1348-a5e3-ae45420ea57d</IDValue>
    </ReferencedID>
  </ReferencedIDs>
</LogRecordBody>
</LogRecordElement>

```

### Log Record Record Description

- 1 The LogRecordElement contains a single log record, corresponding to a single system event.
- 2 A UUID value that uniquely identifies this event. This ID must be the same wherever this event appears (*i.e.*, if the event appears in more than one report, the ID will be the same.)
- 3 The time and date at which the event occurred.
- 4 The sequence number of this event in the report.
- 5 The event *Class* (*e.g.*, *Security*.)
- 6 The event *Type* (*e.g.*, *Key*.)
- 7 Gives the UUID most closely associated with the content element that was being handled when the event occurred.
- 8 The SHA-1 message digest of the Header element in the record that preceded this one in the report.
- 9 The SHA-1 message digest of the Body element contained within the same parent LogRecordElement element.
- 10 Describes the event *Sub-type* (*e.g.*, *KDMKeysReceived*.)
- 11 A list of parameters which augment the event sub-type.
- 12 If an exception (an error) occurred during the procedure that generated the event, this element will contain a list of tokens which describe the error.
- 13 A list of important identifiers that existed in the procedure context when the event occurred.

### 5.3.1.3. Log Report Signature

An XML Signature is used to create a tamper-proof encoding. The signature is made over the contents of the RecordAuthData element as shown in the following example.

#### Example 5.3. Log Report Signature Example

```

<LogRecordSignature> 1
  <HeaderPlacement>stop</HeaderPlacement>
  <SequenceLength>2</SequenceLength>
  <RecordAuthData Id="ID_RecordAuthData"> 2
    <RecordHeaderHash>SG93IE1hbnkgTW9yZSBSZXZpc2lvbnM</RecordHeaderHash> 3
    <SignerCertInfo> 4
      <ds:X509IssuerName>CN=DistCo-ca,OU=DistCo-ra,O=DistCo-ra,
dnQualifier=vnqteTcB2Gji\+lH123sxxgOqvWE</ds:X509IssuerName>
      <ds:X509SerialNumber>16580</ds:X509SerialNumber>
    </SignerCertInfo>
  </RecordAuthData>
  <Signature> 5
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#ID_RecordAuthData">
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>VGhpncyBvbmx5IHRvb2sgdHdvIHllYXJz</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>

```

```

<ds:SignatureValue>
Vqe6MS0pHovkfqhHlkt/NNEI1GGchCW/EyqxOccSenuzNQc63qL+VIQoIJCcwgne0i/w/8bIgjfb
PrsOW5M3z1R0eAZc7tt6f7q50taNmC+O2wfATVXqEE8KC32qO//NQHuOL6bLLH+12oqgR5fS/mlI
/wpn8s/pAtGA9lAXDRp03EVOvzwq0m9AjzOxIbgzGg6AIY0airJlgecTlqcdblzGQjB8lpr3ctlp
ECchubtSCqh+frRn4CZc4ZRMLhjmax/zwhIG4EximCEKbwaz7DwN8zvlyoPUzut9ik7X0EYfRIlv
F3piQoLeeFcFrkfNwYyyhTX8iHTO4Cz8YfGNyw==</ds:SignatureValue>
<ds:KeyInfo>
  <ds:X509Data>
    <ds:X509IssuerSerial>
      <ds:X509IssuerName>Sample Issuer Name</ds:X509IssuerName>
      <ds:X509SerialNumber>1234567</ds:X509SerialNumber>
    </ds:X509IssuerSerial>
    <!-- X509 certificate value as block of Base64 encoded characters, -->
    <!-- truncated for brevity -->
    <ds:X509Certificate>
      QSBdZXJ0aWZpY2F0ZSB3b3VsZCBizSBsb25nZXIgdGhhbiB0aGlz</ds:X509Certificate>
    </ds:X509Data>
    <ds:X509Data>
      <ds:X509IssuerSerial>
        <ds:X509IssuerName>Sample Issuer Name 2</ds:X509IssuerName>
        <ds:X509SerialNumber>123456789</ds:X509SerialNumber>
      </ds:X509IssuerSerial>
      <!-- X509 certificate value as block of Base64 encoded characters, -->
      <!-- truncated for brevity -->
      <ds:X509Certificate>TG9uZ2VyIHRoYW4gdGhpcyB0b28sIGZvciBzdXJl</ds:X509Certificate>
    </ds:X509Data>
  </ds:KeyInfo>
</Signature>
</LogRecordSignature>

```

### Log Record Signature Description

- 1 The LogRecordSignature contains the signature of a log record.
- 2 The RecordAuthData element is the content that is actually signed for the signature. This element is identified for the signature processor by the Id attribute value.
- 3 A message digest value calculated over the sibling Header element.
- 4 This information identifies the creator of the XML Signature (the document's signer.).
- 5 A standard XML Signature element.

#### 5.3.1.4. Log Report Signature Validation

XML Signatures on log reports can be checked using the procedure in Section 3.1.3.

## 5.3.2. Event Log Operations

### 5.3.2.1. Log Structure

#### Objective

- Verify that the Log Report is an XML document and that it validates against the XML schemas defined in [SMPTE-430-4] and [SMPTE-433M].
- Verify that the Log Report contains urn:uuid values as specified in [RFC-4122].

#### Procedures

1. Using the **schema-check** software utility, validate the XML file structure against the XML schemas in [SMPTE-430-4] and [SMPTE-433M]. Failure to correctly validate is cause to fail this test. For more information on schema validation see Section 1.3: Conventions and Practices.

```
$ schema-check <input-file> smpte-433.xsd smpte-430-4.xsd
schema validation successful
$
```

2. Supply the filename of the Log Report file as an argument to the **uuid\_check.py** software utility. Examine the output for error messages that identify expected UUID values that do not conform to the format specified in [RFC-4122]. One or more occurrences is cause to fail this test.

```
$ uuid_check.py <input-file>
all UUIDs conform to RFC-4122
$
```

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.2
[RFC-4122]	
[SMPTE-430-4]	
[SMPTE-433M]	

Test Equipment
schema-check
uuid_check.py

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 5.3.2.2. Log Records for Multiple SPBs

#### Objective

*Only applies to an IMB Security Manager (SM) which has one or more remote SPBs in the auditorium suite.*

Verify that in the case of reports covering multiple SPBs, records consist of a separately identifiable sequence for each reporting SPB.

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the records consist of a separately identifiable sequence for each SPB in the suite.
4. Failure to correctly identify the originating SPB for any event (log records) shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.1

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)
KDM for 2K StEM

### 5.3.2.3. Log Sequence Numbers

#### Objective

Verify that the security manager (SM) maintains a secure and persistent counter to provide a unique sequential number to each log record it creates. Verify that this sequence number appears in the Header node of each log record in a report.

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the header in each record contains a SerialNumber value that is one greater than the value in the previous record.
4. Failure to correctly sequence log records in a report shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.4

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)
KDM for 2K StEM

### 5.3.2.4. Log Collection by the SM

#### Objective

*Only applies to an IMB Security Manager (SM).*

Verify that the SM collects log information from all remote SPBs in the suite it enables, at the earliest equipment idle time. Verify that TLS sessions are not terminated prior to collection of all remote SPB log data and that in no event more than 24 hours will pass between the recording of log data by a remote SPB and the collection of such data by the IMB SM (in a powered state).

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Leave the system idle for 20 minutes, then extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the report contains records identified as having been generated by each remote SPB enabled for the payout.
4. Failure to correctly collect and report remote SPB event records shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.1

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)
KDM for 2K StEM

### 5.3.2.5. General Log System Failure

#### Objective

Only applies to an IMB Security Manager (SM).

- Verify that the system requires that the secure logging subsystem is operating as a prerequisite to playback.
- Verify that the SM will not enable for playback any remote SPB for which it has not collected, or cannot collect log records, or where there is any indication that the remote SPB will not record and report log records as required.

#### Procedures

1. Configure the Test Subject (an IMB) to use the **asm-responder** simulator program instead of a normal projector. Configure the ASM simulator to respond to event requests with a busy response code.
2. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
3. Allow 20 minutes to elapse.
4. Attempt to set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM*.
5. If the SM allows payout, this shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.10

Test Equipment
asm-responder

Test Material
DCI 2K StEM Test Sequence (Encrypted)

## 5.3.3. SM Proxy of Log Events

### 5.3.3.1. SM Proxy of Log Events

#### Objective

Verify that an SM can proxy (for an SPB) log records which contain an unknown class or type of information.

#### Procedures

1. Configure the Test Subject to use the **asm-responder** program as a remote SPB (a virtual LDB). Configure the **asm-responder** to return the set of proprietary test messages.

```
$ asm-responder (...standard options...) \  
  --preload-log-event Prop1.xml \  
  --preload-log-event Prop2.xml \  
  --preload-log-event Prop3.xml
```

Note: The "proprietary" test messages are valid [SMPTE-430-4] log records that contain class or type information not defined in a standard document.

2. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP). Playback may be stopped any time after two minutes.
3. Wait no more than 20 minutes after playout stops to allow the SM to gather the remote SPB logs.
4. Extract a security log report from the Test Subject.
5. Examine the log report using a text editor. Verify that the log contains at least one record each of Class Debug, Type Info, Event Subtype Prop1, Prop2, and Prop3.
6. Verify that each event identified in the previous step has correctly formatted parameters as defined in Appendix D.
7. Failure to correctly record each of the Operations events shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
asm-responder
Text Editor



Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 5.3.3.2. SM Proxy of Security Operations Events

#### Objective

Verify that an SM can proxy (for an SPB) log records which contain correctly coded Security Operations events per [SMPTE-430-5].

#### Procedures

1. Configure the Test Subject to use the **asm-responder** program as a remote SPB (a virtual LDB). Configure the **asm-responder** to return the set of Operations test messages.

```
$ asm-responder (...standard options...) \  

  --preload-log-event SPBOpen.xml \  

  --preload-log-event SPBClose.xml \  

  --preload-log-event SPBMarriage.xml \  

  --preload-log-event SPBDivorce.xml \  

  --preload-log-event SPBShutdown.xml \  

  --preload-log-event SPBStartup.xml \  

  --preload-log-event SPBClockAdjust.xml \  

  --preload-log-event SPBSoftware.xml
```

2. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP). Playback may be stopped any time after two minutes.
3. Wait no more than 20 minutes after playout stops to allow the SM to gather the remote SPB logs.
4. Extract a security log report from the Test Subject.
5. Examine the log report using a text editor. Verify that the log contains at least one record each of Class Security, Type Operations, Event Subtypes SPBOpen, SPBClose, SPBMarriage, SPBDivorce, SPBShutdown, SPBStartup, SPBClockAdjust, and SPBSoftware.
6. Verify that each event identified in the previous step has correctly formatted parameters as defined in [SMPTE-430-5].
7. Failure to correctly record each of the Operations events shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
asm-responder
Text Editor

<b>Test Material</b>
DCI 2K StEM Test Sequence (Encrypted)

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Discontinued May 8 2009

### 5.3.3.3. SM Proxy of Security ASM Events

#### Objective

Verify that an SM can proxy (for an SPB) log records which contain correctly coded Security ASM events per [SMPTE-430-5].

#### Procedures

1. Configure the Test Subject to use the **asm-responder** program as a remote SPB (a virtual LDB). Configure the **asm-responder** to return the set of ASM test messages.

```
$ asm-responder (...standard options...) \  

  --preload-log-event SPBOpen.xml \  

  --preload-log-event SPBClose.xml \  

  --preload-log-event SPBMarriage.xml \  

  --preload-log-event SPBDivorce.xml \  

  --preload-log-event SPBShutdown.xml \  

  --preload-log-event SPBStartup.xml \  

  --preload-log-event SPBClockAdjust.xml \  

  --preload-log-event SPBSoftware.xml
```

2. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP). Playback may be stopped any time after two minutes.
3. Wait no more than 20 minutes after playout stops to allow the SM to gather the remote SPB logs.
4. Extract a security log report from the Test Subject.
5. Examine the log report using a text editor. Verify that the log contains at least one record each of Class Security, Type ASM, Event Subtypes LinkOpened, LinkClosed, LinkException, LogTransfer, and KeyTransfer.
6. Verify that each event identified in the previous step has correctly formatted parameters as defined in [SMPTE-430-5].
7. Failure to correctly record each of the ASM events shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
asm-responder
Text Editor

<b>Test Material</b>
DCI 2K StEM Test Sequence (Encrypted)

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Discontinued May 8 2009

## 5.4. Security Log Events

Secure Processing Blocks (SPB) are required to record Security Log Events (defined in [SMPTE 430-5]) upon the occurrence of certain operational states. The procedures in this section should cause the Test Subject to record the respective events.

### 5.4.1. Payout, Validation and Key Events

#### 5.4.1.1. FrameSequencePlayed Event

##### Objective

Verify that the SM can produce log records which contain correctly coded FrameSequencePlayed events per [SMPTE-430-5].

##### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Payout, Event Subtype FrameSequencePlayed.
4. Verify that the FrameSequencePlayed record has correctly formatted parameters as defined in [SMPTE-430-5].
5. Failure to correctly record a FrameSequencePlayed event shall be cause to fail this test.

##### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)

### 5.4.1.2. CPLStart Event

#### Objective

Verify that the SM can produce log records which contain correctly coded CPLStart events per [SMPTE-430-5].

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Playout, Event Subtype CPLStart.
4. Verify that the CPLStart record has correctly formatted parameters as defined in [SMPTE-430-5].
5. Failure to correctly record a CPLStart event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)

### 5.4.1.3. CPLend Event

#### Objective

Verify that the SM can produce log records which contain correctly coded CPLend events per [SMPTE-430-5].

#### Procedures

1. Set up and play a show completely using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Playout, Event Subtype CPLend.
4. Verify that the CPLend record has correctly formatted parameters as defined in [SMPTE-430-5].
5. Failure to correctly record a CPLend event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 5.4.1.4. PlayoutComplete Event

#### Objective

Verify that the SM can produce log records which contain correctly coded `PlayoutComplete` events per [SMPTE-430-5].

#### Procedures

1. Set up and play a show completely using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the log contains at least one record of Class `Security`, Type `Playout`, Event Subtype `PlayoutComplete`.
4. Verify that the `PlayoutComplete` record has correctly formatted parameters as defined in [SMPTE-430-5].
5. Failure to correctly record a `PlayoutComplete` event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)



### 5.4.1.5. CPLCheck Event

#### Objective

Verify that the SM can produce log records which contain correctly coded CPLCheck events per [SMPTE-430-5].

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Validation, Event Subtype CPLCheck.
4. Verify that the CPLCheck record has correctly formatted parameters as defined in [SMPTE-430-5].
5. Failure to correctly record a CPLCheck event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)

### 5.4.1.6. KDMKeysReceived Event

#### Objective

Verify that the SM can produce log records which contain correctly coded KDMKeysReceived events per [SMPTE-430-5].

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Key, Event Subtype KDMKeysReceived.
4. Verify that the KDMKeysReceived record has correctly formatted parameters as defined in [SMPTE-430-5].
5. Failure to correctly record a KDMKeysReceived event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 5.4.1.7. KDMDeleted Event

#### Objective

Verify that the SM can produce log records which contain correctly coded KDMDeleted events per [SMPTE-430-5].

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Cause the system to remove all KDMs.
3. Extract a security log report from the Test Subject.
4. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Key, Event Subtype KDMDeleted.
5. Verify that the KDMDeleted record has correctly formatted parameters as defined in [SMPTE-430-5].
6. Failure to correctly record a KDMDeleted event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)

## 5.4.2. ASM and Operations Events

### 5.4.2.1. LinkOpened Event

#### Objective

Verify that the SM can produce log records which contain correctly coded LinkOpened events per [SMPTE-430-5].

#### Procedures

1. Power up the Test Subject and associated suite equipment (which should include at least one remote SPB), wait for the system to become idle.
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type ASM, Event Subtype LinkOpened.
4. Verify that the LinkOpened record has correctly formatted parameters as defined in [SMPTE-430-5].
5. Failure to correctly record a LinkOpened event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

### 5.4.2.2. LinkClosed Event

#### Objective

Verify that the SM can produce log records which contain correctly coded LinkClosed events per [SMPTE-430-5].

#### Procedures

1. From the idle state, power down the Test Subject and associated suite equipment (which should include at least one remote SPB).
2. Power up the Test Subject, wait for the system to become idle.
3. Extract a security log report from the Test Subject.
4. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type ASM, Event Subtype LinkClosed.
5. Verify that the LinkClosed record has correctly formatted parameters as defined in [SMPTE-430-5].
6. Failure to correctly record a LinkClosed event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 5.4.2.3. LinkException Event

#### Objective

Verify that the SM can produce log records which contain correctly coded LinkException events per [SMPTE-430-5].

#### Procedures

1. From the idle state, disconnect the ASM communication channel ( *i.e.*, the Ethernet) to a remote SPB.
2. Set up and attempt to play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (this step should fail, *i.e.*, the playout should not occur.)
3. Extract a security log report from the Test Subject.
4. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type ASM, Event Subtype LinkException.
5. Verify that the LinkException record has correctly formatted parameters as defined in [SMPTE-430-5].
6. Failure to correctly record a LinkException event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)

### 5.4.2.4. LogTransfer Event

#### Objective

Verify that the SM can produce log records which contain correctly coded LogTransfer events per [SMPTE-430-5].

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Wait no more than 20 minutes after playout stops to allow the SM to gather the remote SPB logs.
3. Extract a security log report from the Test Subject.
4. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type ASM, Event Subtype LogTransfer.
5. Verify that the LogTransfer record has correctly formatted parameters as defined in [SMPTE-430-5].
6. Failure to correctly record a LogTransfer event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)

### 5.4.2.5. KeyTransfer Event

#### Objective

Verify that the SM can produce log records which contain correctly coded KeyTransfer events per [SMPTE-430-5].

#### Procedures

1. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
2. Extract a security log report from the Test Subject.
3. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type ASM, Event Subtype KeyTransfer.
4. Verify that the KeyTransfer record has correctly formatted parameters as defined in [SMPTE-430-5].
5. Failure to correctly record a KeyTransfer event shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)



### 5.4.2.6. SPBStartup SPBShutdown Events

#### Objective

Verify that the SM can produce log records which contain correctly coded SPBStartup and SPBShutdown events per [SMPTE-430-5].

#### Procedures

1. Power up the Test Subject and associated suite equipment, wait for the system to become idle.
2. Power down the Test Subject and associated suite equipment, power up, wait for the system to become idle.
3. Extract a security log report from the Test Subject.
4. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Operations, Event Subtypes SPBStartup and SPBShutdown.
5. Verify that the SPBStartup and SPBShutdown records have correctly formatted parameters as defined in [SMPTE-430-5].
6. Failure to correctly record SPBStartup and SPBShutdown shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

### 5.4.2.7. SPBOpen SPBClose Events

#### Objective

Verify that the SM can produce log records which contain correctly coded SPBOpen and SPBClose events per [SMPTE-430-5].

#### Procedures

1. Power up the Test Subject and associated suite equipment (which should include at least one SPB with a secure perimeter access door), wait for the system to become idle.
2. Open a secure perimeter access door. Wait one minute, close the access door.
3. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP).
4. Extract a security log report from the Test Subject.
5. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Operations, Event Subtypes SPBOpen and SPBClose.
6. Verify that the SPBOpen and SPBClose records have correctly formatted parameters as defined in [SMPTE-430-5].
7. Failure to correctly record SPBOpen and SPBClose shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)
KDM for 2K StEM

### 5.4.2.8. SPBClockadjust Event

#### Objective

Verify that, for SMs which support clock adjustment, the SM can produce log records which contain correctly coded SPBClockadjust events per [SMPTE-430-5].

#### Procedures

1. Power up the Test Subject and associated suite equipment, wait for the system to become idle.
2. Using the manufacturer's documented procedure, adjust the clock of the Test Subject.
3. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP). Playout may be stopped after one minute.
4. Extract a security log report from the Test Subject.
5. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Operations, Event Subtypes SPBClockadjust.
6. Verify that the SPBClockadjust records have correctly formatted parameters as defined in [SMPTE-430-5].
7. Failure to correctly record SPBClockadjust shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)
KDM for 2K StEM

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 5.4.2.9. SPBMarriage and SPBDivorce Events

#### Objective

Verify that the SM can produce log records which contain correctly coded SPBMarriage and SPBDivorce events per [SMPTE-430-5].

#### Procedures

1. Power up the Test Subject and associated suite equipment, wait for the system to become idle.
2. Using the manufacturer's documented procedure, divorce the Test Subject from its companion device.
3. Using the manufacturer's documented procedure, remarry the Test Subject with its companion device.
4. Return the Test Subject to the idle state (reboot after software installation is acceptable).
5. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP). Playout may be stopped after one minute.
6. Extract a security log report from the Test Subject.
7. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Operations, Event Subtypes SPBMarriage and SPBDivorce.
8. Verify that the SPBMarriage and SPBDivorce records have correctly formatted parameters as defined in [SMPTE-430-5].
9. Failure to correctly record SPBMarriage and SPBDivorce shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-4] [SMPTE-430-5]	9.4.6.3.7

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted) KDM for 2K StEM

### 5.4.2.10. SPBSoftware Event

#### Objective

Verify that the SM can produce log records which contain correctly coded Software events per [SMPTE-430-5].

#### Procedures

1. Power up the Test Subject and associated suite equipment, wait for the system to become idle.
2. Using the manufacturer's documented procedure, perform a software installation on the Test Subject.
3. Return the Test Subject to the idle state (reboot after software installation is acceptable).
4. Set up and play a show using the DCP and KDM contained in *DCI 2K StEM Test Sequence (Encrypted)* and *KDM for 2K StEM* (valid DCP). Playout may be stopped after one minute.
5. Extract a security log report from the Test Subject.
6. Examine the log report using a text editor. Verify that the log contains at least one record of Class Security, Type Operations, Event Subtypes Software.
7. Verify that the Software records have correctly formatted parameters as defined in [SMPTE-430-5].
8. Failure to correctly record Software shall be cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7
[SMPTE-430-4]	
[SMPTE-430-5]	

Test Equipment
Text Editor

Test Material
DCI 2K StEM Test Sequence (Encrypted)
KDM for 2K StEM

**Page Intentionally Left Blank**

# Chapter 6. Media Block

The Media Block (MD) is a Type 1 SPB comprising a Security Manager (SM) and the Media Decryptors (MD) for all essence types, plus, as required, Forensic Marker (FM) for image or sound, a Link Encryptor (LE) and a Timed Text rendering engine (alpha-channel overlay).

## 6.1. Security Manager (SM)

### 6.1.1. Image Integrity Checking

#### Objective

- Verify that the SM detects and logs playback restarts.
- Verify that the SM processes image essence integrity pack metadata, to detect and log deviations from the intended image file (Track File ID) and the intended frame sequence.
- Record whether the SM performs a real-time check of the image frame hash (HMAC).

Note that an image frame hash (HMAC) check is encouraged to be performed by the SM, but optional.

#### Procedures

1. Load composition *DCI 2K StEM Test Sequence (Encrypted)*. Start Playback, stop playback and restart playback. Examine the log file to verify that the restart was logged, and that the log entry was signed by the SM.
2. Playback the composition *DCI Malformed Test 1: Picture with Frame-out-of-order error*, in which two frames are swapped. Record whether, after playback, the occurrence of these two swapped image frames is logged, and that the log entry was signed by the SM.
3. Playback the composition *DCI DCP With an incorrect picture file (Wrong ID)*, in which the image track file has been replaced with another one with a different Track File ID. Record whether, after playback, the occurrence of these replaced track files is logged and that the log entry was signed by the SM.

Failure of any of the above conditions is cause to fail this test.

4. Playback the composition *DCI Malformed Test 9: Picture with HMAC error in MXF Track File*, in which the HMAC value for one of the image frames has been malformed. Record whether after playback, the occurrence of the malformed HMAC is logged, and that the log entry was signed by the SM.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

Test Equipment
DCI Projector

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Used May 8 2009

<b>Test Material</b>
DCI 2K StEM Test Sequence (Encrypted)
DCI Malformed Test 1: Picture with Frame-out-of-order error
DCI DCP With an incorrect picture file (Wrong ID)
DCI Malformed Test 9: Picture with HMAC error in MXF Track File



## 6.1.2. Sound Integrity Checking

### Objective

- Verify that the SM processes sound essence integrity pack metadata, to detect and log deviations from the intended sound file (Track File ID) and the intended frame sequence.
- Verify that the SM performs a real-time check of the sound frame hash (HMAC).

### Procedures

1. Playback the composition *DCI Malformed Test 2: Sound with Frame-out-of-order error*, in which two sound frames are swapped. Record whether after playback, the occurrence of these two swapped sound frames is logged, and that the log entry was signed by the SM.
2. Playback the composition *DCI DCP With an incorrect sound file (Wrong ID)*, in which the sound track file has been replaced by another one with a different Track File ID. Record whether after playback, the occurrence of these replaced track files is logged, and that the log entry was signed by the SM.
3. Playback the composition *DCI Malformed Test 10: Sound with HMAC error in MXF Track File*, in which the HMAC value for one of the sound frames has been malformed. Record whether after playback, the occurrence of the malformed HMAC is logged, and that the log entry was signed by the SM.

Failure of any of the above conditions is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

Test Material
DCI DCP With an incorrect sound file (Wrong ID)
DCI Malformed Test 2: Sound with Frame-out-of-order error
DCI Malformed Test 10: Sound with HMAC error in MXF Track File

### 6.1.3. Restriction of Keying to Monitored Link Decryptors

#### Objective

Verify that the SM issues Link Encryption keys only to Link Decryptors (LDB or LD/LE) in remote SPBs that show no sign of improper operation or attack (as determined by means of the QuerySPB message).

#### Procedures

Using the ASM responder simulator configured to always set the Security Alert condition of the response to the QuerySPB command, make sure that the SM does not issue LEKeyLoad Request (set key) messages to the ASM device. If the LEKeyLoad Request message is received, this is cause to fail this test.

```
$ asm-responder (... standard options ...) \  
--respond-with-queryspb-type SecurityAlert
```

Note: In architectures that support multiple projectors and/or the use of LD/LE image processing SPB devices, this test should be carried out for each additional device. In addition to the QuerySPB response, the key usage qualification criteria include assuring ONLY these devices appear on the TDL.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

Test Equipment
asm-responder

## 6.1.4. Restriction of Keying to MD Type

### Objective

Verify that keys are issued only to MDs matching the key type as specified in the KDM per [SMPTE-430-1].

### Procedures

1. Load the KDM *KDM with mismatched keytype*, which contains a valid decryption key for image, but the Key Type is mismatched.
2. Attempt to play the composition and record the result. Verify that the composition cannot be played. Failure to meet this requirement is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5
[SMPTE-430-1]	

Test Material
KDM with mismatched keytype

## 6.1.5. Restriction of Keying to valid CPLs

### Objective

Verify that the SM validates CPLs and logs results as a prerequisite to preparing the suite for the associated composition playback.

### Procedures

1. Supply the CPL *DCI CPL with an Invalid Signature* to the SM. Verify that the SM rejects the CPL. If the SM accepts the CPL, this is cause to fail this test.
2. Examine the log file to verify that the rejection of the CPL was logged, and that the log entry was signed by the SM. Failure to meet these requirements is cause to fail this test.
3. Attempt to start playback and verify that it is not possible. If playback starts, this is cause to fail this test.

Note: Certain SMS implementations may check the CPL for validity as a prerequisite to handing the CPL to the SM. Depending on the way the SMS is implemented, it may be necessary to use a special mode (e.g., by clicking a CPL validation error ignore button) to deliver the CPL to the SM. In case the SMS prevents supplying invalid CPLs to the SM, the manufacturer must provide an SMS with modified software that allows this kind of operation.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

Test Material
DCI CPL with an Invalid Signature

## 6.1.6. Remote SPB Integrity Monitoring

### Objective

Verify that the SM continuously monitors and logs integrity status of all Remote SPBs, to detect attacks during normal operation, by issuing the QuerySPB command at least every 30 seconds.

### Procedures

For all remote SPBs, perform the following three tests:

First test:

1. Set up the system, load the DCP *DCI 2K StEM Test Sequence* and playback the composition.
2. Disconnect the 100BaseT control network cable between the IMB SPB and the remote SPB.
3. Record whether the composition continues to play.
4. Examine the logs and verify that the failed QuerySPB request is logged and signed by the SM. The failure of this event to be recorded or correctly signed is cause to fail this test.

Second test (only possible for projector SPBs):

1. Set up the system, load the DCP *DCI 2K StEM Test Sequence* and prepare the projection system for playout (i.e., establish TLS sessions)
2. Wait one minute, open the projector access door for one minute, then close the access door for another minute.
3. Verify that the SM logs and signs an "open door" security alert for the minute the door was open via the expected QuerySPB commands. The failure of this event to be recorded or correctly signed is cause to fail this test.

Third test (only possible for projector SPBs):

1. Set up the system, load the DCP *DCI 2K StEM Test Sequence* and playback the composition.
2. Wait one minute, open the projector access door for one minute, then close the access door for another minute.
3. Record whether playback terminates when the access door is opened and that the playback does not automatically resume when the projector door is closed. Failure to meet either of these requirements is cause to fail this test.
4. Verify that the SM logs and signs an "open door" security alert for the minute the door was open via the expected QuerySPB commands. The failure of this event to be recorded or correctly signed is cause to fail this test.

Note: This test does not fully cover the requirements as described in the Objective. Using the above functional test, it is only verified that the SM continuously queries the SPB (or at least monitors the connection), but not that SPB integrity status is actually monitored. This is difficult to verify functionally, as integrity failures are not triggered easily for all SPB types without destruction of the SPB. It is therefore preferred to execute this test non-functionally by careful source code analysis in cooperation with the designers.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

<b>Test Material</b>
DCI 2K StEM Test Sequence

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Revised May 8 2009

## 6.1.7. SPB Integrity Fault Consequences

### Objective

Verify that the SM responds to SPB substitutions since the previous authentication and/or ATM QuerySPB Command by terminating and re-establishing:

1. TLS sessions (and re-authenticating the suite) and
2. Suite playability conditions (KDM prerequisites, SPB queries and key loads).

### Procedures

With cooperation of the designers of the system and the necessary block diagrams, schematics and source codes given, verify that in case SPB substitution since the previous authentication and/or ATM status query is detected, the SM:

1. Prevents (or stops, if in progress) playback,
2. Purges all keys (content and link encryption keys),
3. Terminates and re-establishes the TLS sessions, and
4. Re-establishes suite playability conditions such as KDM prerequisites, SPB queries ("QuerySPB") and key loads.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

## 6.2. Link Encryption (LE)

*This section is only applicable for systems that use Link Encryption.*

### 6.2.1. LDB Trust

#### Objective

Verify that for playback of content that is not encrypted (therefore no KDM or TDL for this content exists) the SM automatically assumes "trust" in the LDB and projector SPBs for purposes of keying the LDB and enabling playback.

#### Procedures

1. Remove and restore power to the equipment containing the IMB. This forces fresh TLS sessions to the remote SPBs and ensures that fresh LE keys are generated.
2. Set up a show containing the composition *DCI 2K StEM Test Sequence*. This DCP is unencrypted and therefore has no associated KDM with a TDL.
3. Playback the show and verify that the composition plays without problems. Failure to successfully and completely playback is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.4

Test Material
DCI 2K StEM Test Sequence



## 6.2.2. Multiple LE Operation

### Objective

- Verify that the SM enables multiple LE operation only when the SM receives a KDM whose TDL contains only the identities of the remote SPBs identified during TLS authentication.
- Verify that the SM shall not support the use of more than one image processor SPB (LD/LE) for any LDB/projector configuration
- Verify that the SM shall key the multiple LDB/projector configuration using different LE keys.

### Procedures

1. By carrying out the following steps, setup a system comprising the IMB under test and four (4) ASM responders, two (2) which will simulate projector SPB type 2s and two (2) which will simulate their respective LDBs.
  - a. Create a file containing the signer certificate chain by concatenating the certificates PRJ-chain-b1-root.pem, PRJ-chain-b1-signer1.pem and PRJ-chain-b1-signer2.pem, e.g:

```
$ cat PRJ-chain-b1-root.pem PRJ-chain-b1-signer1.pem \
    PRJ-chain-b1-signer2.pem > foo.pem
```

- b. Initialize each ASM responder with the appropriate values taken from Table 6.1 below. The IP addresses for the responders may be replaced by any available addresses in the same subnet as the IMB is configured. An example is shown below:

```
$ asm-responder --bind-address 192.168.1.100 --certificate-file chain-b1-spb2-1.pem \
    --certificate-chain-file foo.pem --private-key-file rsa-key-spb2-1.pem --tls-only
```

2. Power up the part of the system that contains the IMB and observe that all the TLS connections become established. From this point forward record all LEKeyLoad events which are announced by the LDB ASM responders.
3. Load and playback the test materials *KDM for multiple LDs, 2 LDBs*.
4. Perform a second playback of the test materials from the step above.
5. Verify that at least two (2) LE keys have been sent to each LDB ASM responder and that every key is different. Failure to meet both requirements is cause to fail this test.
6. Load and attempt to playback the test materials *KDM for multiple LDs, 2 LD/LE, 1 LDB*.

**Table 6.1. ASM responder parameters**

Responder Role	IP Address	Certificate File	Private Key File	Additional Arguments
LDB #1	192.168.1.100	chain-b1-ldb-1.pem	rsa-key-ldb-1.pem	

Responder Role	IP Address	Certificate File	Private Key File	Additional Arguments
SPB2 #1	192.168.1.101	chain-b1-spb-1.pem	rsa-key-spb2-1.pem	--tls-only
LDB #2	192.168.1.102	chain-b1-ldb-2.pem	rsa-key-ldb-2.pem	
SPB2 #2	192.168.1.103	chain-b1-spb-2.pem	rsa-key-spb2-2.pem	--tls-only

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.4.1

Test Equipment
asm-responder

Test Material
KDM for multiple LDs, 2 LDBs
KDM for multiple LDs, 2 LD/LE, 1 LDB

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.0 released May 8 2009

## 6.2.3. LE Key Usage

### Objective

Verify that a fresh Link Encryption key is used for each movie showing.

### Procedures

Using an ASM responder simulator to display the ASM messages sent to the LDB, perform the following procedures:

1. Start the ASM responder simulator:

```
$ asm-responder (... standard options ...)
```

2. Setup and play the composition *DCI 2K StEM Test Sequence (Encrypted)*.
3. Record the LD key delivered to the ASM responder simulator.
4. Repeat Step 1.
5. Record the LD key delivered to the ASM responder simulator. If a new LD key is not delivered this is cause to fail this test.
6. Verify that the LD key recorded in Step 2 is different from that recorded in Step 4. Failure to meet this requirement is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.4

Test Equipment
asm-responder

Test Material
DCI 2K StEM Test Sequence (Encrypted)

## 6.2.4. IMB Link Encryption

### Objective

In the case that the IMB is external to the Projector SPB (or the IMB and Projector are not an integrated subsystem), verify that Link Encryption is applied. Verify that Link Encryption is applied for both plaintext and ciphertext compositions.

### Procedures

1. Attach a suitable monitor to the interface that connect the Projector and the Media Block (MB) without disrupting the Projector-MB connection (i.e., "tap" the connection; this may require soldering a special test adapter).
2. Setup and play a show using *DCI 2K StEM Test Sequence (Encrypted)* This test material contains an encrypted composition.
3. Verify that the image is displayed properly on the projection screen.
4. Verify that only a scrambled signal is seen on the "tapped" monitor. A non-scrambled image on the monitor is cause to fail this test.
5. Setup and play a show using *DCI 2K StEM Test Sequence* This test material contains a plaintext composition.
6. Verify that the image is displayed properly on the projection screen.
7. Verify that only a scrambled signal is seen on the monitor. A non-scrambled image on the monitor is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.2.2.10, 8.4.2, 8.4.3.1

Test Equipment
Dual-Link Monitor
Bridge Tap Connector
DCI Projector

Test Material
DCI 2K StEM Test Sequence (Encrypted)
DCI 2K StEM Test Sequence

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 2009

## 6.3. Clocks and Time

This section describes general requirements concerning the time awareness of the projection system and its individual components. All procedures are applicable to the Security Manager, with the notable exception of section 6.3.2, which is applicable to all SPBs of type 1.

### 6.3.1. Clock Adjustment

#### Objective

- Verify that in order to maintain synchronization between auditoriums, exhibitors are able to adjust a SM's clock offset a maximum of +/- six minutes within any calendar year.
- Verify that the SM clock offset time adjustments are logged events.

#### Procedures

1. Verify that there is a possibility to adjust the clock.
2. Attempt to adjust the clocks more than +/- six minutes and verify that this fails.
3. Verify that the above adjustments are logged.

Note: the above test procedure does not fully cover the requirements of the objective, as the above test involves the SMS as an intermediate between the tester and the SM under test. Having the SMS as an intermediate, however, means that it cannot be tested that it is actually the SM who restricts clock adjustments. (The apparent restriction also could be done "client side" in the SMS). For fully covering the objective, the objective must be verified in cooperation with the manufacturer, with access to the necessary source codes.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.7

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 6.3.2. Clock Battery

#### Objective

- Verify that all Type 1 SPBs maintain their time-awareness under both powered and unpowered conditions.
- Verify that the Type 1 SPB clock's battery has a life of 5 years at a minimum.
- Verify that the Type 1 SPB clock's battery is changeable without losing track of proper time.
- Verify that time stamping of log entries is not interrupted during a battery change process.

#### Procedures

For all SPBs of type 1:

1. Verify that there is a battery for clock buffering, and, in the case of an IMB, verify that this battery can be changed without losing track of proper time.
2. In unpowered condition, make the SPB generate some log entries and note down the time when this is done relative to an external reference clock (however, do not use tampering for generating log entries, as this renders the SPB inoperable). Then change the clock buffer batteries of the SPB. Verify that after powering the system up again, log entries with the correct times (relative to the external reference clock) have been written. (For LDBs this is already tested in Section 7.4.3: LDB Time-Awareness ).
3. Use special measuring equipment (current meter) to determine how much current is consumed by the buffering circuit of the clock. This can be done by wiring the current meter in series with the buffer battery. Do this measurement for both powered and unpowered conditions and note down the higher one of both values. From the battery data sheets, determine how much energy the battery can store. From those two values, calculate the maximum time the battery can power the clock buffer. Verify that this is longer than 5 years.

Note: the above test does not fully cover the objective, as the minimum battery life depends on several factors, one being ambient temperature. Describing such tests in detail is beyond the scope of this test plan. It is suggested to execute test procedure 3 in cooperation with the manufacturer.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.7

Test Equipment
DMM
Digital Clock

### 6.3.3. Clock Resolution

#### Objective

Verify that the IMB SM clock has a resolution to one second.

#### Procedures

1. Setup and playback a show containing the composition *128 Reel Composition, "A" Series (Encrypted)*. This DCP contains 128 reels of encrypted essence, each with a duration of one (1) second.
2. Examine the log records produced by the above playback. If the time stamps of the log entries are recorded to one (1) second resolution, it can be deduced that the SM clock has a resolution of at least one second. Failure to meet this requirement is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.7

## 6.4. Forensic Marking (FM)

### 6.4.1. FM Application Constraints

#### Objective

- Verify that FM is not applied to non-encrypted audio or image content.
- Verify that FM is not applied to Track Files that are not encrypted in case portions of a composition are encrypted and other portions are not.

#### Procedures

1. Playback the DCP *FM Constraints* and present the reproduced image and sound to the appropriate Forensic Marking (FM) detector. This package has a CPL that selects between encrypted and plaintext, image and sound track files in a specific order.
2. Verify that the FM detectors report the following status for the presentation:
  - a. The first three minutes of the presentation should indicate no image FM, and no sound FM.
  - b. The section between three minutes and six minutes should indicate image FM present, but no sound FM.
  - c. The section between six minutes and nine minutes should indicate no image FM, and sound FM present.
  - d. The section between nine minutes until the end of the presentation should indicate both image FM and sound FM.

Any discrepancy between the expected and reported FM states is cause to fail this test.

Note: the equipment manufacturer is required to provide a suitable FM decoder (i.e., software and hardware).

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.2

Test Equipment
FM Detector

Test Material
FM Constraints



## 6.4.2. Granularity of FM Control

### Objective

- Verify that "No FM mark" states are capable of being independently controlled, for audio and image, via appropriate use of the ForensicMarkFlagList element of the KDM for audio or image Track Files.
- Verify that the ForensicMarkFlagList element of the KDM and thus the "no FM mark" state applies to the entire CPL/composition, according to the associated KDM.
- Verify that the "no FM mark" state does not apply to any other composition, even if the other composition is part of the same showing (i.e., same Show Playlist).

### Procedures

1. Build a show playlist out of the following four test materials: *KDM with no Forensic Marking enabled*, *KDM with Image Forensic Marking enabled*, *KDM with Audio Forensic Marking enabled*, *KDM for 2K StEM*.
2. Play back the Show, and present the reproduced image and sound to the appropriate Forensic Marking (FM) detector.
3. Verify that the FM detectors report the following status for the presentation:
  - a. *KDM with no Forensic Marking enabled*: no image FM and no audio FM for the whole composition
  - b. *KDM with Image Forensic Marking enabled*: image FM present, but no audio FM, for the whole composition
  - c. *KDM with Audio Forensic Marking enabled*: no image FM, but audio FM present, for the whole composition
  - d. *KDM for 2K StEM*: image FM and audio FM present for the whole composition

Any discrepancy between the expected and reported FM states is cause to fail this test.

Note: the equipment manufacturer is required to provide a suitable FM decoder (i.e., software and hardware).

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.2

Test Equipment
FM Detector

Test Material
KDM with no Forensic Marking enabled
KDM with Image Forensic Marking enabled
KDM with Audio Forensic Marking enabled
KDM for 2K StEM

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### 6.4.3. FM Payload

#### Objective

- Verify that the Forensic Marking data payload is a minimum of 35 bits, and contains both time stamp and location data.
- Verify that every 15 minutes, 24 hours per day, 366 days/year are time stamped (will repeat annually).
- Verify that 16 bits (enough values for all the possible 35,136 time stamps) are allocated for the time stamp.
- Verify that 19 bits (524,288 possible locations/serial numbers) are allocated for location (serial number) information.
- Verify that all 35 bits are included in each five minute segment.
- Verify that recovery is possible with a 30-minute content sample for positive identification.

#### Procedures

1. Setup and play a show using the DCP *KDM for 2K StEM*.
2. Play a section of 30 minutes in length and use appropriate image and audio FM detectors to extract the data payload of the Forensic Marking.
3. Verify that the Forensic Marking decoder indicates that a "positive identification" has been made.
4. Verify that at least the following data is contained within both image and audio:
  - a. 16 bit time stamp.
  - b. 19 bit location ID.
5. Verify that two or three sequential time stamps have been recovered during the 30 minute content sample.

To verify that all possible time stamps are generated would prove impractical, as testing would need to continue for a full calendar year. Design review is necessary to verify this assertion.

An assessment of whether all 35 bits are included in each 5 minute segments may be made if the Forensic Marking decoder is capable of providing data before "positive identification" is confirmed. This does not fully cover the objective, however, which can only be verified by design review.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.1.1

Test Equipment
FM Decoder

Test Material
KDM for 2K StEM

## 6.5. Image Reproduction

### 6.5.1. Playback of Image Only Material

#### Objective

Verify that all projection systems are capable of playing back content that consists of image only, i.e., has no corresponding audio or other track.

#### Procedures

Playback the DCP *DCI NIST Frame no sound files*. This package comprises image only. Verify that the image is displayed correctly. Failure to display the image is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.3.1.3

Test Material
DCI NIST Frame no sound files

## 6.5.2. Decoder Requirements

### Objective

Verify that the image decoder meets all requirements for JPEG 2000 image decoder presented in [DCI\_DCSS-1-1] Sec. 4.3.2.

### Procedures

1. Verify that the decoder supports both 2K and 4K images, with 24 fps for 4K (decode to 2K or 4K) and 24 or 48 fps for 2K (decode to 2K). To verify this, build a show using *DCI 2K Sync Test (2K@24fps)*, *DCI 2K Sync Test (48fps) (2K@48fps)* and *4K Sync Test (4K@24fps)*. Verify that playback is successful and that image and audio are played back properly.

2. Verify that the output of the decoder conforms to the image specification:

- a. 12 bit X'Y'Z' color

To test for 12 bit color reproduction: Load the test image contained in *DCI Gray Steps*. The image shows a special gray scale stepping pattern for ensuring usage of all 12 bits. Verify that each of the 12 gray scale bars of the pattern is displayed correctly, i.e., all steps should be equal in width; luminance should decrease continuously from one side to the other.

Load the test image contained in *DCI Grayscale Gradient*. The image shows a gray scale gradient test pattern. Verify that no stepping is visible.

- b. Testing for X'Y'Z' color reproduction is already covered by Section 7.5.11: Transfer Function.

- c. One of the following image formats/frame rates:

- i. 4096 x 2160 at 24 fps

- ii. 2048 x 1080 at 24 fps

- iii. 2048 x 1080 at 48 fps

3. Verify that the decoder decodes each color component at 12 bits per sample with equal color/component bandwidth and does not subsample chroma (i.e., does not generate any 4:2:2 signal or similar). This must be verified in cooperation with the manufacturer.

4. Verify that the decoder decodes all data for every frame in a 4K distribution and does not discard data to keep up with peak data rates. Verify that no data (e.g., J2K resolution levels or quality layers) is discarded even for peak data rates (exception: the highest resolution level for 4K images may be discarded where the projector is only 2K capable).

- Play the DCP contained within {TBD:}, which is the 4K STEM test material with a counter burned in as subtitles that is incremented by one with every frame. The projected image must be filmed with a suitable camera and then be viewed in slow motion to verify that no counter numbers are skipped.
- For 2K Projectors: Play the DCP contained in {TBD:}, which contains 2K J2K images with the highest resolution level (2K) modified to contain specially crafted patterns. Verify that the projected image contains a clearly visible, regular pattern that does not change over time (i.e., no flickering or similar).

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 release M38 2009

- For 4K Projectors: Play the DCP contained in {TBD:}, which contains 4K J2K images with highest resolution level (4K) modified to contain specially crafted patterns. Verify that the projected image contains a clearly visible, regular pattern that does not change over time (i.e., no flickering or similar).
5. Together with the manufacturer, verify that the 9/7 wavelet transform and the Inverse Irreversible Color Transformation (ICT) are implemented with at least 16 bit fixed point precision. In case the hardware description or software source code for the decoding algorithm is available, this can be checked by careful analysis.
  6. The manufacturer must guarantee in written form that the decoder does not need to be updated once it has been deployed.

**Supporting Materials**

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	4.3.2

Test Equipment
48 fps Camera

Test Material
DCI 2K Sync Test
DCI 2K Sync Test (48fps)
4K Sync Test
DCI Gray Steps
DCI Grayscale Gradient

## 6.6. Audio Reproduction

### 6.6.1. Digital Audio Interfaces

#### Objective

Verify that the Media Block has a digital audio output interface with the capacity for delivering 16 channels of digital audio at 24-bit 48 kHz or 96 kHz, and follows the [AES3-2003] recommended practice for serial transmission format for two-channel linearly represented digital audio data.

#### Procedures

1. Playback 16 channel id composition and verify correct output from all 16 channels. Failure to confirm reproduction on any channel is cause to fail this test.
2. Playback 48kHz composition and verify:
  - a. 48kHz AES3 signal at outputs.
  - b. Pink noise bandwidth to 22kHz.
  - c. 24 active bits on analyzer.

Failure to confirm above conditions a, b and c, is cause to fail this test.

3. Playback 96kHz composition and verify:
  - a. 96kHz AES3 signal at outputs.
  - b. Pink noise bandwidth to 44kHz.
  - c. 24 active bits on analyzer.

Failure to confirm above conditions a, b and c, is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[AES3-2003]	
[DCI-DCSS-1-1]	7.5.6.1, 7.5.6.2

Test Equipment
AES3 Audio Analyzer

Test Material
DCI 0-16 Numbered Channel Identification
DCI NIST Frame with Pink Noise

<b>Test Material</b>
DCI NIST Frame with Pink Noise (96 kHz)

Version 1.0 ARCHIVE ONLY  
NEWEST VERSION AVAILABLE  
Discontinued May 8 2009

## 6.6.2. Audio Sample Rate Conversion

### Objective

Verify that the system has the capability of performing Sample Rate Conversion (SRC) when needed.

### Procedures

1. Play back the DCP *DCI NIST Frame with 1 kHz tone (-20 dB fs, 96kHz)*. Enable SRC on the system, select an output rate of 48kHz. With an AES analyzer, confirm that each of the AES-3 outputs are producing an AES signal with a 48kHz sample rate. Any other measured output sample rate is cause to fail this test.
2. Play back the DCP *DCI NIST Frame with 1 kHz tone (-20 dB fs)*. Enable SRC on the system, select an output rate of 96kHz. With an AES analyzer, confirm that each of the AES-3 outputs are producing an AES signal with a 96kHz sample rate. Any other measured output sample rate is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	3.3.2.3

Test Equipment
AES3 Audio Analyzer

Test Material
DCI NIST Frame with 1 kHz tone (-20 dB fs)
DCI NIST Frame with 1 kHz tone (-20 dB fs, 96kHz)



### 6.6.3. Audio Delay Setup

#### Objective

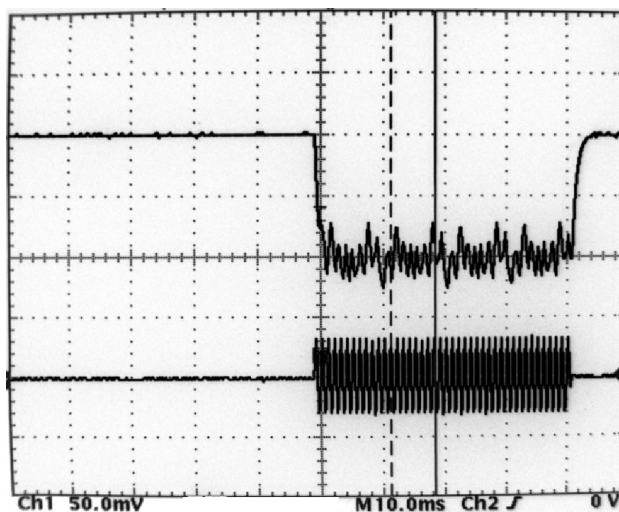
Verify that the system provides a method for adjusting the delay of the audio signal relative to the image. It must be possible to offset audio +/-200 mS in 10 msec increments.

#### Procedures

1. Connect channel 1 of the oscilloscope to the analog center channel output of the sound equipment.
2. Connect channel 2 of the oscilloscope to a photodiode that is placed in front of the projection screen, where the flashing rectangle is located.
3. Perform the following steps:
  - a. Playback the DCP contained in *DCI 2K Sync Test*. This Package contains short beeps (one frame in length) and a white flashing rectangle at the bottom of the screen, synchronized to the beeps.
  - b. Measure the delay between the light pulse and the audio pulse. This will depend on a combination of many factors such as the image processing delay of the display device, sound processing delay in the sound equipment, and digital signal transmission delays (buffering of data). Record the timing with zero offset applied to the unit under test. Use this nominal figure as the reference point for the following step.
  - c. Verify that the timing can be adjusted +/- 200 msec, in 10 msec steps, and that the delay that is observed at the oscilloscope varies accordingly. Failure to meet this requirement is cause to fail this test.
4. Repeat the above test, but this time for 48 fps (use the DCP *DCI 2K Sync Test (48fps)*). Record the results obtained.

The image below shows what a typical measurement is expected to look like. The upper trace shows the light output of the projector, measured by means of the photo diode. The photo diode signal is shown inverted, i.e., low means high light output. The lower trace shows the analog center channel output of the Media Block after D/A conversion from the AES-EBU signal.

**Figure 6.1. Audio Delay Timing**



Warning: the optical flashes generated during this test can cause epileptic seizures with persons who have abnormal predispositions for photosensitive epilepsy. Persons with such predispositions should not be present during testing.

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

**Supporting Materials**

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.4.1.8

Test Equipment
Oscilloscope
Photodiode

Test Material
DCI 2K Sync Test
DCI 2K Sync Test (48fps)

## 6.6.4. Click Free Splicing of Audio Track Files

### Objective

Verify that the playback system allows click-free splicing of the audio track files.

### Procedures

Playback the DCP *DCI Malformed Test 3: Sound Splice Tests*. This package has a CPL that causes a 400 Hz sine wave tone to be spliced repeatedly, in a way that will ensure an amount of phase discontinuity at the splice point. Record any occurrence of audible snaps, crackles or pops, the reproduced audio should have no evidence of unpleasant artifacts at the splice points.

Note: Playback of this test must be done in a properly equipped and set up movie theater, at reference level, i.e., fader setting 7.0 for Dolby and compatibles or fader setting 0 dB for Sony and compatibles. A single channel of pink noise at -20dBFS should produce a Sound Pressure Level (SPL) of 85dBc, from any of the front loudspeakers, at the monitoring position. Monitoring by means of smaller monitor boxes or headphones is not sufficient.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	5.3.1.6

Test Equipment
Sound System

Test Material
DCI Malformed Test 3: Sound Splice Tests

## 6.7. Timed Text Reproduction

### 6.7.1. Media Block Overlay

#### Objective

In the case that the Media Block implements an alpha channel overlay module, a subpicture renderer (a module that converts the subpicture file into a baseband image file with an alpha channel) and a Timed Text renderer (a module that converts Timed Text data into a baseband image file with an alpha channel), verify that assets are rendered and displayed correctly by the system.

#### Procedures

1. Using a digital cinema projector that does not provide an internal subtitle rendering capability (or one in which subtitle rendering capability is disabled), load and play the composition *DCI 2K Sync test with Subtitles*.
2. Verify that the timed text and subpicture instances update synchronously with the burned-in text.
3. Failure to verify correct synchronization shall be cause to fail the test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.5.4.2.5, 7.5.4.2.6, 7.5.4.2.7

Test Equipment
DCI Projector

Test Material
DCI 2K Sync test with Subtitles

## 6.7.2. Timed Text Synchronization

### Objective

Verify that timed text and subpicture essence is correctly synchronized to sound and picture essence.

### Procedures

1. Load and play the composition *DCI 2K Sync test with Subtitles*.
2. Verify that the timed text and subpicture instances update synchronously with the burned-in text.
3. Failure to verify correct synchronization shall be cause to fail the test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	3.4.2.5, 3.4.3.8

Test Material
DCI 2K Sync test with Subtitles

### 6.7.3. Support for Multiple Captions

#### Objective

Verify that the timed text reproduction system supports multiple lines of text.

#### Procedures

1. Load and play the composition *Multi-line Subtitle Test*.
2. Verify that the timed text instances contain multiple lines of text.
3. Failure to correctly display multiple lines of text shall be cause to fail the test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	3.4.3.7

Test Material
Multi-line Subtitle Test

## 6.7.4. Default Timed Text Font

### Objective

Verify that a timed-text rendering system provides a default font to be used in the case where no font files are supplied with the DCP.

### Procedures

1. Load and play the composition *Multi-line Subtitle Test w/o Integral Font*.
2. Verify that the timed text instances contain multiple lines of text.
3. Failure to verify correct synchronization shall be cause to fail the test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	3.4.3.4

Test Material
Multi-line Subtitle Test w/o Integral Font

**Page Intentionally Left Blank**



---

# Chapter 7. Projector

The Projector is a Type 2 SPB comprising a light processing system, including electronic and optical components, and a companion SPB. The projector may be stand-alone, in which case the companion SPB will be a Link Decryptor (LD), or else the companion SPB will be an Image Media Block (IMB). The projector may include a Timed Text rendering engine (alpha-channel overlay).

## 7.1. Projector Test Environment for Image Measurements

When making image measurements on a Test Subject, the following environmental conditions must exist:

The Test Subject (projector) must be turned on (including the lamp) and allowed to thermally stabilize for 20 to 30 minutes prior to all measurements. The projector's color management system must be configured such that incoming code values are interpreted in accordance with [SMPTE-428-1].

Stray light on the screen must be minimized. The room lights in screening rooms must be turned off, with the exception of the minimal lighting provided for working or safety reasons. For a theatrical environment room, the room lights must be the normal theatrical lighting environment. The ambient light level of a mastering environment reflected by the screen must be less than 0.01 cd/m<sup>2</sup> (.0029 ft-L), that of a theatrical environment less than 0.03 cd/m<sup>2</sup> (.01 ft-L). The use of black non-reflective surfaces with recessed lighting is encouraged. Safety regulations and the placement of exit lights or access lights can result in a higher ambient light level.

The screen must be non-specular and equally reflective over the entire visible spectrum. The screen must have variable black masking, adjustable to tightly frame the projected image (at a minimum, this must include the 1.85:1 and 2.39:1 image formats).

All image parameters must be measured off of the screen from the center of the normal seating area in an exhibition theater. All measurements must be done according to [SMPTE-431-1], [SMPTE-431-2] and [SMPTE-431-3].

## 7.2. SPB Type 2

### 7.2.1. Projector Physical Protection

#### Objective

Verify that SPB type 2 protection is provided for the Companion SPB type 1 and its plaintext image interfaces within the projector.

#### Procedures

1. By physical examination, determine the physical perimeter that provides the SPB Type 2 protection for the Projector.
2. Locate and for each of any removable access covers and/or doors of the SPB Type 2, record whether they are protected by either:
  - a. Pick resistant mechanical locks employing physical or logical keys, or
  - b. Tamper-evident seals (e.g. evidence tape or holographic seals).

The absence of one or more sorts of protection listed above on any of the access covers or doors comprising part of the SPB Type 2 perimeter is cause to fail this test.
3. By physical examination, locate the Companion SPB of the Projector (either an Image Media Block (IMB) or a Link Decryptor Block (LDB)).
4. Verify that the Companion SPB is entirely enclosed within, or mechanically connected to, the Projector's SPB Type 2 enclosure. Failure to meet this requirement is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.2.2, 9.4.3.6.1, 9.5.2.4

## 7.2.2. Projector Access Door

### Objective

Verify that the projector SPB implements a "projector SPB access door open" event signal to the companion SPB. Verify that playback is not permitted and terminates if the projector SPB access door is open.

### Procedures

Carefully examine the projector SPB for access doors. If any are found, execute the following steps for each one and record the results.

1. Playback the DCP *DCI 2K StEM Test Sequence*.
2. Open the projector access door and observe that playback terminates. If playback does not terminate, this is cause to fail this test.
3. Attempt to start playback with the door open. If playback starts, this is cause to fail this test.
4. Examine the logs from the projector's companion SPB and verify that a "door open" event was created for each time a door was opened. If any log record is missing, this is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.1

Test Material
DCI 2K StEM Test Sequence

## 7.2.3. SPB2 Requirements

### Objective

Verify that for SPB Type 2, [FIPS-140-2] level 3 requirements are followed for "Cryptographic Module Ports and Interfaces", with TLS security as defined in these specifications providing input/output logical separation protection if TLS is used for projector authentication. Verify that for SPB Type 2, the operational environment of the secure chip follows [FIPS-140-2] "Limited Operational Environment".

### Procedures

Together with the manufacturer and with access to the necessary schematics and component specifications, verify the SPB type 2 under test has been implemented as described by [FIPS-140-2] in the following areas:

1. Conformance to [FIPS-140-2] level 3 requirements for "Cryptographic Module Ports and Interfaces" (with TLS security as defined in these specifications providing input/output logical separation protection if TLS is used for projector authentication).
2. Conformance of the operational environment of the secure chip to the [FIPS-140-2] "Limited Operational Environment".

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.4
[FIPS-140-2]	

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 7.2.4. SPB2 Secure Silicon Requirements

### Objective

Verify that SPB type 2 secure silicon:

- Is tamper resistant and of the type that zeroizes Critical Security Parameters (CSPs) prior to exposure upon physical attack.
- Is tamper responsive.
- If the secure silicon is of the type that is subject to FIPS 140-2 Level 3, Area 5 (physical security), per the requirements of Section 9.5.2.5 of the DCI Specification v1.1, it must meet this requirement.

### Procedures

To pass this test the manufacturer must present a data sheet that guarantees FIPS 140-2 Level 3 compliance of the SPB type 2 secure silicon device under test in the following areas:

1. Resistance to physical (i.e., tamper resistance) and logical attacks.
2. Tamper responsiveness.
3. Zeroization of Critical Security Parameters (such as RSA private keys) during a physical attack, prior to exposure of secrets.
4. If of the type fitted with covers and/or doors, that the secure silicon will detect tampering and is tamper responsive.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.1, 9.5.2.2, 9.5.2.5

## 7.2.5. SPB2 Tamper Evidence

### Objective

Verify that SPB type 2 is tamper evident. (Maintenance of the SPB type 2 (projector) is permitted for non-security components accessible via maintenance openings)

### Procedures

By careful visual and physical examination, verify that the SPB type 2 under test is tamper evident. This means that all tampering, no matter how carefully executed, shall leave visible traces on the hardware. Examination should include attempting to access the SPB in several ways. Maintenance of the projector SPB is permitted for non-security components via maintenance access openings.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.1

## 7.2.6. SPB2 Secure Silicon Field Replacement

### Objective

Verify that the secure silicon device, contained within a SPB type 2, is not field serviceable (may be field replaceable).  
Verify that it is not accessible during normal SPB type 2 operation or non-security-related servicing.

### Procedures

By careful optical and physical examination, verify that the secure silicon device contained within a SPB type 2

1. is not field serviceable (but may be field replaceable), i.e., there are no provisions for direct access to the SPB type 2 secure silicon circuitry.
2. is not accessible during normal SPB type 2 operation or non-security-related servicing, i.e., is mounted in a special compartment separated from areas accessible during operations or normal servicing.
3. is clearly described in the system manual as not being field serviceable

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.3

## 7.2.7. Systems Without Electronic Marriage

### Objective

Verify that in the configuration of a permanently married companion SPB (IMB or LDB), the companion SPB is not field replaceable and requires the projector SPB and companion SPB system to both be replaced in the event of an SPB failure.

### Procedures

Verify that the companion SPB type 1 (IMB if no link encryption is used or LDB if link encryption is used) is not field-replaceable. Careful optical and physical inspection is necessary for this. Any deviation from these requirements is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.6



Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 7.2.8. Electronic Marriage Break Key Retaining

### Objective

Verify that breaking the marriage between the projector and its companion SPB (LDB or IMB) does not zeroize the projector SPB type 2 long term identity keys (RSA private keys).

### Procedures

1. Using an ASM requester simulator, initiate a TLS session with the projector SPB and capture the certificate supplied by the projector in PEM encoded format.

```
$ asm-requester (... standard options ...)
```

2. Using the procedure illustrated in Section 2.1.11, record the public key thumbprint of the certificate captured in the above step.
3. Intentionally break the marriage and remarry the systems (this may require support by the manufacturer). Make sure that no RSA keys are uploaded into the system during the remarriage process.
4. Verify that, after remarriage, the system is able to re-establish a TLS session. Failure to establish a TLS session after remarriage is cause to fail this test.
5. Using the same procedure as described in steps 1 and 2, verify that the public key in the certificate supplied by the projector is the same as before the remarriage. Mismatching public key thumbprints are cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.1

Test Equipment
asm-requester

## 7.3. Companion SPB Type 1

### 7.3.1. Projector Companion SPB Location

#### Objective

Verify that the Projector's companion SPB (LDB or IMB) is physically inside of, or otherwise mechanically connected to, the Projector SPB.

#### Procedures

By means of an optical inspection, verify that the projector's companion SPB (LDB or IMB) is physically inside or otherwise mechanically connected to the projector SPB. Deviation from this requirement is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.1

## 7.3.2. Companion SPBs with Electronic Marriage

### Objective

This test only applies to field replaceable companion SPBs (IMB or LDB) that implement electronic marriage functions.

- Verify that as part of the installation, or reinstallation, (i.e., mechanical connection to the projector and electrical initiation) an electrical and logical marriage of the companion SPB (IMB or LDB) with the projector SPB is performed.
- Verify that upon initiation of the marriage a "SPBMarriage" log record is written (per [SMPTE-430-5]) and that the record contains all required data.
- Verify that upon break of the marriage a "SPBDivorce" log record is written (per [SMPTE-430-5]) and that the record contains all required data.

### Procedures

1. Verify system is functional prior to breaking the marriage. This can be achieved by loading and successfully playing the composition *DCI 2K StEM Test Sequence*.
2. Power down the system, locate the field-replaceable companion SPB (IMB or LDB), break the marriage by disconnecting and/or removing the SPB.
3. Replace and reconnect the companion SPB, power up the system, examine the logs and verify that a "SPBDivorce" log record has been written. Absence of this entry is cause to fail this test.
4. Verify the following are contained in the SPBDivorce record:
  - a. The `DeviceSourceID` element contains the Certificate Thumbprint of the companion SPB.
  - b. The `DeviceConnectedID` element contains the Certificate Thumbprint of the projector SPB2.
  - c. The log entry contains an `AuthID` record.Failure to meet requirements a, b and c above is cause to fail this test.
5. Setup a show with composition from Step 1. Verify that the system does not play the composition. Failure to meet this requirement is cause to fail this test.
6. Perform the marriage installation procedure and repeat Step 1 to verify that the system is now capable of payout. Failure to meet this requirement is cause to fail this test.
7. Examine the logs and verify that a "SPBMarriage" log entry has been written. Absence of this entry is cause to fail this test.
8. Verify the following are contained in the SPBMarriage record:
  - a. The `DeviceSourceID` element contains the Certificate Thumbprint of the companion SPB.
  - b. The `DeviceConnectedID` element contains the Certificate Thumbprint of the projector SPB2.
  - c. The log entry contains an `AuthID` record.

Failure to meet requirements a, b and c above is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.1, 9.4.3.6.2, 9.4.3.6.3
[SMPTE-430-5]	

Test Material
DCI 2K StEM Test Sequence

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

### 7.3.3. Companion SPB Marriage Break Key Retaining

#### Objective

Verify that breaking the marriage between the companion SPB (LDB or IMB) and the projector SPB does not zeroize the companion SPB's long term identity keys (RSA private keys).

#### Procedures

(Only applies to systems that implement an Electronic Marriage, i.e., those that have field replaceable LDBs or IMBs.)

1. Using an ASM requester simulator, initiate a TLS session with the companion SPB (LDB or IMB) and capture the certificate supplied by the companion SPB in PEM encoded format.
2. Using the procedure illustrated in Section 2.1.11, record the public key thumbprint of the certificate captured in the above step.
3. Intentionally break the marriage and remarry the systems (this may require support by the manufacturer). Make sure that no RSA keys are uploaded into the system during the remarriage process.
4. Verify that, after remarriage, the system is able to re-establish a TLS session. Failure to establish a TLS session after remarriage is cause to fail this test.
5. Using the same procedure as described in steps 1 and 2, verify that the public key in the certificate supplied by the companion SPB upon initialization is the same as before the remarriage. Mismatching public key thumbprints are cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.2, 9.4.3.6.3

Test Equipment
asm-requester

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 7.4. Link Decryptor Block

The Link Decryptor Block (LDB) is a Type 1 SPB that is used to receive encrypted signals into a Type 2 SPB companion device (such as a projector).

### 7.4.1. LDB without Electronic Marriage

#### Objective

Verify that projector implementations with permanently married companion SPBs do not permit field replacement of the IMB or LDB as appropriate according to which function the companion SPB performs. In the event of an SPB failure the projector SPB and companion SPB are required to be replaced as a pair.

#### Procedures

1. Locate the companion SPB within the projector SPB type 2 and verify that it is not possible to be replaced in the field. Deviation from this requirement is cause to fail this test.
2. Verify and record that the system manual clearly states that field replacement of the companion SPB is not allowed.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.6

## 7.4.2. LDB TLS Session Constraints

### Objective

Verify that LDBs do not establish security communications with more than one SM at a time.

### Procedures

1. Configure a system comprising the SMS/IMB/LDB under test and an additional SMS/IMB.
2. Use a computer running network monitoring software to monitor the security network and confirm the opening of a TLS session between one of the SMs and the LDB under test.
3. When the other SM attempts to open a TLS session with the LDB, confirm that the LDB either rejects connections to the TCP port or closes the connection immediately after establishment. Deviation from this behavior is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.2

Test Equipment
Network Analyzer
D.U.T. Twin

### 7.4.3. LDB Time-Awareness

#### Objective

Verify that the LDB contains a UTC reference clock which is backed up by battery and operative for time stamping log events under powered and unpowered conditions.

#### Procedures

1. Make sure the system has been fully operational (i.e., initialized) at least once before this test.
2. Note down the system clock relative to an external digital reference clock, thus making calculation of the relative clock offset possible.
3. Power off the system.
4. Open the projector SPB 2 door and note down the external reference clock time at which this happened.
5. Power up the system again.
6. Examine the log records and verify that the opened SPB door has been logged at the right time. This can be verified because the offset of the internal clock relative to the external reference clock is known, so the expected internal clock time of the open door event can be calculated. An incorrect time stamp is cause to fail this test.
7. While powered, open the projector door and verify that an according log entry is written with the correct time. An incorrect time stamp is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.2

Test Equipment
Accurate Real-Time Clock



## 7.4.4. LDB ASM Conformity

### Objective

Verify that the LDB responds to SM QuerySPB requests and other ASMs as necessary to conform to requirements specified in [SMPTE-430-6] and the additional requirements specified in the DCSS Section 9.4.5.3.2.

### Procedures

1. Using the ASM requester simulator, start a TLS session with the LDB and for each of the General Purpose and Link Encryption ASM Commands listed in [SMPTE 430-6], issue the command. Record the response and the time elapsed between issuing the command and receiving the response.

```
$ asm-requester ( ... standard options ... ) \  
--messagetype <messagetype>
```

2. Examine the results obtained from Step 1 for cases where the response is not received within two seconds, or at all. Any such cases are cause to fail this test.
3. Examine the results obtained from Step 1 for cases that contain the ResponderBusy general response element code "3". Any such cases are cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-6]	9.4.3.6.2, 9.4.5.3.2

Test Equipment
asm-requester

## 7.4.5. LDB Key Storage

### Objective

Verify that the LDB accepts and stores LD keys and associated parameters provided by the SM. Verify that the LDB has the capacity to store at least 16 key/parameter sets.

### Procedures

1. Using an ASM requester simulator, initiate a TLS session with the LDB and issue an LEKeyPurgeAll command.

```
$ asm-requester (... standard options ...) --messagetype LEKeyPurgeAll
```

2. Using an ASM requester simulator, issue an LEKeyQueryAll command. The response should indicate an empty LE key list. A non-empty list shall be cause to fail this test.

```
$ asm-requester (... standard options ...) --messagetype LEKeyQueryAll
list size: 0
```

3. Using an ASM requester simulator, issue sixteen (16) LEKeyLoad commands. Verify that the LE key list contains sixteen keys by executing an LEKeyQueryAll command. An LE key list size other than sixteen shall be cause to fail this test.

```
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 1
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 2
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 3
...
$ asm-requester (... standard options ...) --messagetype LEKeyQueryAll
list size: 16
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.2

Test Equipment
asm-requester

## 7.4.6. LDB Key Purging

### Objective

Verify that the LDB purges LD keys under the following conditions:

1. upon expiration of the SM-designated validity period
2. upon receipt of a "LEKeyPurgeAll" command from the SM

### Procedures

1. Using an ASM requester simulator, initiate a TLS session with the LDB and issue an LEKeyPurgeAll command.

```
$ asm-requester (... standard options ...) --messagetype LEKeyPurgeAll
```

2. Using an ASM requester simulator, issue an LEKeyQueryAll command. The response should indicate an empty LE key list. A non-empty list shall be cause to fail this test.

```
$ asm-requester (... standard options ...) --messagetype LEKeyQueryAll
list size: 0
```

3. Using an ASM requester simulator, issue an LEKeyLoad command with a validity period of one minute. Verify that the LE key list contains one (1) key by executing an LEKeyQueryAll command. An LE key list size other than one shall be cause to fail this test.

```
$ asm-requester (... standard options ...) --messagetype LEKeyLoad \
--messagetype-id 4
$ asm-requester (... standard options ...) --messagetype LEKeyQueryAll
list size: 1
```

4. Wait one minute. Using an ASM requester simulator, issue an LEKeyQueryAll command. The response should indicate an LE key list with one key. An empty LE key list or a list size greater than one shall be cause to fail this test.

```
$ asm-requester (... standard options ...) --messagetype LEKeyQueryAll
list size: 0
```

5. Using an ASM requester simulator, issue six (6) LEKeyLoad commands. Verify that the LE key list contains six keys by executing an LEKeyQueryAll command. An LE key list size other than six shall be cause to fail this test.

```
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 1
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 2
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 3
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 4
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 5
$ asm-requester (... standard options ...) --messagetype LEKeyLoad --messagetype-id 6
```

```
$ asm-requester (... standard options ...) --messagetype LEKeyQueryAll
list size: 6
```

6. Using an ASM requester simulator, execute an LEKeyPurgeAll command.

```
$ asm-requester (... standard options ...) --messagetype LEKeyPurgeAll
```

7. Using an ASM requester simulator, issue an LEKeyQueryAll command. The response should indicate an empty LE key list. A non-empty list shall be cause to fail this test.

```
$ asm-requester (... standard options ...) --messagetype LEKeyQueryAll
list size: 0
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.2

Test Equipment
asm-requester

## 7.4.7. LDB Logging

### Objective

Verify that the LDB records security event data for logging under both powered and unpowered conditions.

### Procedures

1. Identify one or more events which may occur when the Test Subject is not powered ( *e.g.*, SPBDivorce).
2. With the Test Subject in the powered-down state, perform the actions identified in the previous step.
3. Power up the Test Subject and extract all security log messages using the ASM requester program.
4. Examine the log messages using a text editor. Verify that the log messages reflect the actions taken.
5. Verify that the log records have correctly formatted parameters as defined in [SMPTE-430-5].
6. Failure to correctly record events in a powered-down state shall be cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.2
[SMPTE-430-4]	

Test Equipment
asm-requester

## 7.5. Projector Image Reproduction

### 7.5.1. Projector Overlay

#### Objective

In the case that the Projector implements an alpha channel overlay module, a subpicture renderer (a module that converts the subpicture file into a baseband image file with an alpha channel) and a Timed Text renderer (a module that converts Timed Text data into a baseband image file with an alpha channel), verify that assets are rendered and displayed correctly by the system.

#### Procedures

1. Using a digital cinema server that does not provide an internal subtitle rendering capability (or one in which subtitle rendering capability is disabled), load and play the composition *DCI 2K Sync test with Subtitles*.
2. Verify that the timed text and subpicture instances update synchronously with the burned-in text.
3. Failure to verify correct synchronization shall be cause to fail the test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.4.4

Test Equipment
DCI Server

Test Material
DCI 2K Sync test with Subtitles

## 7.5.2. Projector Lens

### Objective

Record the availability of an automatic lens change solution.

### Procedures

If the projector manufacturer markets a device that allows automatic changeover of a lens, record the information (*e.g.*, model number, lenses supported).

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.2.2.4

### 7.5.3. Projector Pixel Count/Structure

#### Objective

Verify that the sampling structure of the displayed picture array (pixel count of the projector) is equal that of the respective specified image containers (either 4096 x 2160 or 2048 x 1080).

#### Procedures

This test is done using a special "black-and-white binary tree" test picture: on the first line, every pixel alternates between black and white, on the second line every second pixel, on the third line every fourth pixel, and so on. During testing, every pixel block must be visually examined.

Prior to beginning the following procedures, it is necessary to verify that any electronic rescaling of the image is fully disabled. This may include turning off resizing, keystone correction and/or other related processes.

For 2K Projectors:

1. Display the test pattern "*NIST*" *2k Test Pattern* (horizontal pixel test). Verify that the binary tree pixel structure is displayed as expected.
2. Display the test pattern "*NIST*" *2k Test Pattern* (vertical pixel test). Verify that the binary tree pixel structure is displayed as expected.

Deviation from the expected image is cause to fail this test.

For 4K Projectors:

1. Display the test pattern "*NIST*" *4k Test Pattern* (horizontal pixel test). Verify that the binary tree pixel structure is displayed as expected.
2. Display the test pattern "*NIST*" *4k Test Pattern* (vertical pixel test). Verify that the binary tree pixel structure is displayed as expected.

Deviation from the expected image is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.2.2.6, 8.2.2.7

Test Material
"NIST" 2k Test Pattern
"NIST" 4k Test Pattern



## 7.5.4. Projector Spatial Resolution and Frame Rate Conversion

### Objective

Verify that the native display resolution, spatial conversions (where necessary), scaling and frame rates are according to the DCI Specification.

### Procedures

1. Verify that the display has a native resolution of either 4096 x 2160 or 2048 x 1080.
2. In case the native resolution is 4096 x 2160, verify that the projector performs up-conversion of 2048 x 1080 signals, i.e., that the screen is filled as it would be with a 4K image.
3. Verify that all spatial conversions are done at an exact ratio of 2:1 in each axis.
4. In case scaling is used for supporting constant height or constant width projection, visually verify that this scaling does not introduce any visible image artifacts.
5. Verify that image material with frame rates different from the Projection System's native refresh rate is converted by the Projector to the Projector's native refresh rate, i.e., the image is displayed properly.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.2.2.7, 8.2.2.8

## 7.5.5. White Point Luminance and Uniformity

### Objective

- Verify that the peak white luminance measured at the screen center is 48 cd/m<sup>2</sup> as specified according to [SMPTE-431-1].
- Verify that luminance uniformity is as specified according to [SMPTE-431-1].

### Procedures

Set up the projection system as described in [SMPTE-431-1] and display the white field test pattern (X'= 3794, Y'=3960, Z'=3890) contained in the DCP *White Frame Sequence*. Align the lamp house to minimize luminance fall-off from center to corners. The test pattern may already be stored in the Projector for easy setup. In case it is not stored internally it must be provided by means of an external signal source.

1. Adjust the lamp focus or lamp current to a light level of 48 cd/m<sup>2</sup> measured at the screen center. Record the measured light level and any quantitative values of adjustable parameters from the projection system (e.g. lamp power, x/y/x lamp position etc). To meet specification, accuracy must be within tolerances of +/- 3.5 cd/m<sup>2</sup> for review rooms, and +/- 10.2 cd/m<sup>2</sup> for exhibition theaters.
2. Measure the luminance at the four sides. Record the measured light levels. To meet specification, values must be 80% to 90% of center for review rooms and at 75% to 90% for exhibition theaters.
3. In the case of review rooms, measure the luminance at the four corners. Record the measured light levels. To meet specification, values must be 80% to 90% of center for review rooms.

Note: All measurements shall be done as described in [SMPTE-431-1]. Measurement criteria like Projector conditions, measurement locations on the screen, and measurement locations in the auditorium are given in [SMPTE-431-1]. It is expected that this measurement be done in either a review room or movie theater like environment, meaning that the ambient light level is as low as possible and the screen is reasonably large.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-431-1]	8.3.4.3, 8.3.4.4

Test Equipment
Photometer

Test Material
White Frame Sequence

## 7.5.6. White Point Chromaticity and Uniformity

### Objective

- Verify that white point chromaticity is as specified according to [SMPTE-431-1].
- Verify that chromaticity uniformity of the Projection System is as specified according to [SMPTE-431-1].

### Procedures

Display the white field test pattern ( $X'=3794$ ,  $Y'=3960$ ,  $Z'=3890$ ) contained in the DCP *White Frame Sequence*. The test pattern may already be stored in the projector for easy setup. In case it is not stored internally it must be provided by means of an external signal source.

1. Measure the white point chromaticity coordinates at the center of the screen with a spectroradiometer. Record the measured chromaticity values. To meet the specification, the chromaticity values shall be  $x=0.3140$ ,  $y=0.3510$  with a tolerance of  $\pm 0.002$   $x,y$  for review rooms and  $\pm 0.006$   $x,y$  for exhibition theaters.
2. Measure white point chromaticity uniformity by measuring the chromaticity coordinates at the four corners with a spectroradiometer. Record the measured chromaticity values. To meet the specification, the chromaticity values shall be within  $\pm 0.008$   $x,y$  of center for review rooms, and  $\pm 0.015$   $x,y$  of center for exhibition theaters.

Note: All measurements shall be done as described in [SMPTE-431-1]. Measurement criteria like Projector conditions, measurement locations on the screen, and measurement locations in the auditorium are given in [SMPTE 431-1]. It is expected that this measurement be done in either a review room or movie theater like environment, meaning that the ambient light level is as low as possible and the screen is reasonably large.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.3.4.5, 8.3.4.6
[SMPTE-431-1]	

Test Equipment
Spectroradiometer

Test Material
White Frame Sequence

## 7.5.7. Sequential Contrast

### Objective

Verify that the Projector achieves a nominal value sequential contrast of at least 1200:1.

### Procedures

1. Measure the luminance at the center of the screen for the "full black" test pattern contained in the DCP *DCI Black Spacer - 5 seconds*.
2. Measure the luminance at the center of the screen for the "full white" test pattern contained in the DCP *White (White Frame)*.
3. Compute the sequential contrast ratio by dividing the white luminance value by the black luminance value.
4. Record the calculated value. The minimum value required in a theatrical environment is at least 1200.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.3.4.7

Test Equipment
Photometer

Test Material
DCI Black Spacer - 5 seconds
White (White Frame)

## 7.5.8. Intra-frame Contrast

### Objective

Verify that the intra-frame contrast is at least 100:1.

### Procedures

1. Display the checkerboard test pattern *Checkerboard Sequence*.
2. Measure the luminance level at each of the patches in the checkerboard test pattern.
3. Calculate the average value of the luminance of the white patches and divide by the average value of the luminance of the black patches.
4. Record the calculated value. The minimum value required in a theatrical environment is 100.

Note: Sequential contrast is influenced by many factors, such as lens flare, port glass flare, ambient light spilling on the screen and back-reflections from the room itself. Therefore, this measurement should be done in a theater in full operating mode with the port glass and all the painting and seating applied/installed.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.3.4.8
[SMPTE-431-2]	

Test Equipment
Photometer

Test Material
Checkerboard Sequence

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 7.5.9. Grayscale Tracking

### Objective

Using the black-to-white gray and the black-to-dark gray step-scale test patterns, verify that the entire step-scale appears neutral without any visible color non-uniformity or non-monotonic luminance steps in the test pattern.

### Procedures

1. With the Projector powered down or douser closed, use a Spectroradiometer to measure and record the Luminance of the ambient light reflected from the screen.
2. With the Projector powered up, douser open and displaying no image or black code values, use a Spectroradiometer to measure and record the Luminance of the light reflected from the screen.
3. Playback the DCP *DCI White Steps* (black-to-white gray step-scale test pattern).
4. For each of the ten steps of the pattern listed in Table A-2 of [SMPTE-431-2], measure and record the Output Luminance and Chromaticity Coordinates with a Spectroradiometer.
5. The entire step-scale should appear neutral without any visible color non-uniformity or non-monotonic luminance steps in the test pattern. Record the presence of any perceived deviation from a neutral scale.
6. Playback the DCP *DCI Gray Steps* (black-to-dark gray step-scale test pattern).
7. For each of the ten steps of the pattern listed in Table A-3 of [SMPTE-431-2], measure and record the Luminance and Chromaticity Coordinates with a Spectroradiometer.
8. The entire step-scale should appear neutral without any visible color non-uniformity or non-monotonic luminance steps in the test pattern. Record the presence of any perceived deviation from a neutral scale.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.3.4.9
[SMPTE-431-2]	

Test Equipment
Spectroradiometer

Test Material
DCI White Steps
DCI Gray Steps

## 7.5.10. Contouring

### Objective

Verify that no contouring can be observed.

### Procedures

1. Playback the DCP *Contouring*. The package contains a mixture of natural test images, such as horizons, sunsets/sunrises, highlights, and artificial test images with gradients and/or fades.
2. Record any visible contouring artifacts, e.g., appearance of steps or bands where smooth transitions are expected.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.3.4.10
[SMPTE-431-2]	

Test Material
Contouring

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 7.5.11. Transfer Function

### Objective

Verify that the correct encoding transfer function is being used by the projector.

### Procedures

1. With the Projector powered down or douser closed, use a Spectroradiometer to measure and record the Luminance of the ambient light reflected from the screen.
2. With the Projector powered up, douser open and displaying no image or black code values, use a Spectroradiometer to measure and record the Luminance of the light reflected from the screen.
3. Playback the DCP *DCI White Steps* (black-to-white gray step-scale test pattern).
4. For each of the ten steps of the pattern listed in Table A-2 of [SMPTE-RP-431-2-2007], measure and record the Output Luminance and Chromaticity Coordinates with a Spectroradiometer.
5. For each of the measured Output Luminance values, calculate the percentage deviation from the target value and record those results.
6. All the percentage deviations calculated in the last step must fall within +/-5%, if the Projector is to pass the specification for Theatrical Tolerances.
7. Playback the DCP *DCI Gray Steps* (black-to-dark gray step-scale test pattern).
8. For each of the ten steps of the pattern listed in Table A-3 of [SMPTE-RP-431-2-2007], measure and record the Output Luminance and Chromaticity Coordinates with a Spectroradiometer.
9. For each of the measured Output Luminance values, calculate the percentage deviation from the target value and record those results.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.3.4.11
[SMPTE-431-2]	

Test Material
DCI White Steps
DCI Gray Steps



## 7.5.12. Color Accuracy

### Objective

Verify that all colors are accurately reproduced within the tolerances as specified in [SMPTE-RP-431-2-2007].

### Procedures

1. With the Projector powered down or douser closed, use a Spectroradiometer to measure and record the Luminance of the ambient light reflected from the screen.
2. With the Projector powered up, douser open and displaying no image or black code values, use a Spectroradiometer to measure and record the Luminance of the light reflected from the screen.
3. Playback the DCP *Color Accuracy Series*.
4. For each of the twelve color patches listed in Table A-4 of [SMPTE-RP-431-2-2007], measure and record the Output Luminance and Chromaticity Coordinates with a Spectroradiometer.
5. For each of the measured sets of Color Coordinates and Output Luminance values, derive the L\*a\*b\* equivalent values and record them.
6. For each of the reference sets of Color Coordinates and Output Luminance values, derive the L\*a\*b\* equivalent values and record them.
7. Using the formula  $\Delta E^*_{ab} = [(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2]^{1/2}$ , for each pair of values from steps 5 and 6, calculate the delta E\*ab value and record it. Eab\_Calc.py, a tool to perform this calculation, is available in Section C.6.
8. If all calculated delta E\*ab values are  $\leq 4.0$ , the Projector passes the specification for Color Accuracy.

Note: See Annex L of [SMPTE-EG-432-1-2007] for an example of how to convert xyY values to L\*a\*b\* values.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	8.3.4.13
[SMPTE-431-2]	
[SMPTE-432-1]	

Test Equipment
Spectroradiometer eab_calc.py

Test Material
Color Accuracy Series

**Page Intentionally Left Blank**

---

# Chapter 8. Screen Management System

A Screen Management System (SMS) (or Theater Management System (TMS)) is responsible for providing the operator's interface for ingest, scheduling, reporting, etc. In this document the term SMS will be used exclusively, although the same test procedures can apply to a TMS that is able to directly manage a suite of equipment for a screen.

The SMS is not hosted on secure hardware (*i.e.*, it is not required to be within an SPB).

## 8.1. Ingest and Storage

### 8.1.1. Storage System Ingest Interface

#### Objective

Verify that the system provides an interface to the storage system, for DCP ingest, that is Ethernet, 1Gb/s or better, over copper (1000Base-T) or fiber (1000Base-FX), as described in [IEEE 802.3], running the TCP/IP protocol.

#### Procedures

1. Use a computer with the appropriate interface cards, e.g., 1000Base-T copper Ethernet and network analysis tools such as Wireshark installed, to tap the ingest interface.
2. Ingest [2K\_StEM\_sequence\_ct] and verify that the packets can be read by the computer that runs the network analysis tools. Failure to observe the packets contained in the DCP is cause to fail this test.
3. Verify that the data packets read are valid TCP/IP data packets. Use of any other protocol to ingest the DCP is cause to fail this test.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	6.2.3
[IEEE-802-3]	

Test Equipment
Network Analyzer

Test Material
DCI 2K StEM Test Sequence (Encrypted)

## 8.1.2. Storage System Capacity

### Objective

Verify that the storage system available to the SMS has a capacity of at least 1TByte of content.

### Procedures

Verify that the storage system has the capacity to hold at least 1TByte of content. This can be done in three ways:

1. Verify by using the specification of the manufacturer.
2. Examine the capacity of the file system representing the storage system, and verify that there is enough available storage to hold 1 TByte of content data. Use appropriate file system tools to perform this task.
3. Measure the storage capacity by copying 1TByte of content to the storage and verifying that no content has been purged by playing back all content.

If the capacity of the storage system is less than 1TByte, this is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.2.3.11

### 8.1.3. Storage System Redundancy

#### Objective

Verify that the storage system available to the SMS provides redundancy in the case of hard disk failure.

#### Procedures

Verify the existence and functionality of an appropriate RAID configuration by performing the following:

1. Ingest the composition [2K\_StEM\_sequence\_ct] i.e., load it into the storage system.
2. Power down the system.
3. Disconnect one hard drive of the RAID configuration.
4. Re-power the system.
5. Set up and play a show and verify that playback is successful, i.e., playback can be started, is not interrupted and does not show any picture or sound artifacts. Unsuccessful playback is cause to fail this test.
6. Power down the system and reconnect the hard drive that was disconnected in step 3.
7. Repeat the steps 3 to 5 for all other drives contained in the storage system.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.5.3.2

Test Material
DCI 2K StEM Test Sequence (Encrypted)

## 8.1.4. Storage System Performance

### Objective

Verify that the storage system available to the SMS is able to sustain a minimum peak data rate of 307 MBit/sec to allow for non interrupted digital cinema playback.

### Procedures

1. Use the DCP *DCI 2K Sync Test* to build a show. This DCP contains images with minimal image compression, 16 audio channels (each 24 bit per sample, 96 kHz) and subtitles which ensure that the maximum allowable data rate is required for reproduction.
2. Play the show and verify that playback is successful, i.e., playback can be started, is not interrupted and does not show any picture or sound artifacts. Unsuccessful playback is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.5.3.3, 7.5.3.4, 7.5.3.6

Test Material
DCI 2K Sync Test
DCI NIST Frame with silence

## 8.2. Screen Management System

### 8.2.1. Screen Management System

#### Objective

The objective of this test is to verify, for each auditorium, the presence of an SMS which supports the basic functions including the possibility of logging and that this SMS can be operated in one of two modes (local and remote).

#### Procedures

1. Verify that every auditorium has one local SMS (may need manufacturer cooperation).
2. Verify that this SMS can be controlled by the operator, either by local access using a graphical user interface, or by remote access.
3. To verify that the SMS supports basic functions, load *DCI 2K StEM Test Sequence (Encrypted)* and its valid KDM, then setup a show and start and stop playback. Verify proper reaction of the system regarding the functions performed.
4. Verify that the SMS supports additional functions, e.g., DCP and CPL checking and also supports logging of the results of such functions.

Note: The LCD projector control screen present within some DLP projectors is usually not the SMS control screen and thus does not indicate the presence of an SMS.

#### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.2.2, 7.5.8

Test Material
DCI 2K StEM Test Sequence (Encrypted)

## 8.2.2. Show Playlist Creation

### Objective

- Verify that the SMS provides the necessary functions for managing Composition Play Lists (CPLs ) and for assembling them into shows (SPL creation)
- Verify that the SMS allows only authorized persons to build a Show Playlist (SPL).

### Procedures

1. Ingest the DCP contained in *DCI 2K StEM Test Sequence* into the system.
2. Using the system, locate the CPL contained in *DCI 2K StEM Test Sequence*.
3. Create a new Show Play List (SPL) and add *DCI 2K StEM Test Sequence* twice to the show. The two instances of *DCI 2K StEM Test Sequence* are herein referred to as *DCI 2K StEM Test Sequence X* and *DCI 2K StEM Test Sequence Y*.
4. Ingest *DCI 2K StEM Test Sequence (Encrypted)* into the system.
5. Using the system, locate the CPL for *DCI 2K StEM Test Sequence (Encrypted)*.
6. Append the CPL for *DCI 2K StEM Test Sequence (Encrypted)* to the end of the previously assembled show.
7. Move the CPL for *DCI 2K StEM Test Sequence (Encrypted)* in between *DCI 2K StEM Test Sequence X* and *DCI 2K StEM Test Sequence Y*.
8. Ingest *Black (Empty Frame)* and insert it between each of the CPLs in the show.
9. Start playback and verify that the presentation proceeds as expected and the inserted black frames and silence are presented correctly.
10. Ingest *DCI 2K StEM Test Sequence (Encrypted)*.
11. Display all CPLs present in the system.
12. Delete *DCI 2K StEM Test Sequence (Encrypted)*.
13. Wait until playback is completed.
14. Remove *DCI 2K StEM Test Sequence X* from the Show Playlist.
15. Attempt to delete *DCI 2K StEM Test Sequence* from storage. It is expected that the SMS warns the user that this DCP/CPL is part of an SPL and does not allow deletion.
16. Delete the SPL then delete *DCI 2K StEM Test Sequence* and *DCI 2K StEM Test Sequence (Encrypted)*. It is expected that this operation succeeds.
17. Verify that the according content and KDMs have been removed.
18. Verify that the above functions for assembling content into an SPL are executable with an easy to use graphical user interface.



Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1  
 Used May 8 2009

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.2.3.5, 7.2.3.7, 7.3.4, 7.4.1.1, 7.4.1.2, 7.4.1.3, 7.4.1.4, 7.4.1.5, 7.4.1.6

Test Material
DCI 2K StEM Test Sequence
DCI 2K StEM Test Sequence (Encrypted)
Black (Empty Frame)

## 8.2.3. Show Playlist Format

### Objective

Verify that the SMS supports the required Show Playlist Format.

### Procedures

1. Export the Show Playlist (SPL) to external media.
2. Use the software command **schema-check** to verify that the SPL exported in the above step is well formed XML. XML format errors are cause to fail this test. An example is shown below.

```
$ schema-check <input-file>
schema validation successful
```

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.3.2, 7.3.3.1, 7.3.3.2

Test Equipment
schema-check

## 8.2.4. KDM Validity Checks

### Objective

Verify that the SMS supports KDM validity checks in advance of the show.

### Procedures

Verify that the SMS provides a method to check KDM validity and expiry status by performing the following procedures prior to a supposed show:

1. Check the KDM *KDM for 2K StEM* with the appropriate tools the SMS/TMS is providing. The KDM is required to be reported as valid to pass this test procedure.
2. Check the KDM *KDM with invalid XML* with the appropriate tools the SMS/TMS is providing. This KDM is required to be reported as invalid due to invalid file format to pass this test procedure.
3. Check the KDM *KDM that has expired* with the appropriate tools the SMS/TMS is providing. This KDM is required to be reported as expired to pass this test procedure.
4. Check the KDM *KDM with incorrect message digest* with the appropriate tools the SMS/TMS is providing. This KDM is required to be reported as invalid signature to pass this test procedure.
5. Check the KDM *KDM with future validity period* with the appropriate tools the SMS/TMS is providing. This KDM is required to be reported as being valid in the future to pass this test procedure.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.4.1.6

Test Material
KDM for 2K StEM
KDM with invalid XML
KDM that has expired
KDM with incorrect message digest
KDM with future validity period

## 8.2.5. Automation Control and Interfaces

### Objective

Verify that the SMS supports theater automation interface via contact closures (general purpose I/O).

### Procedures

1. Using the GPIO test fixture, verify that the Test Subject can provide contact closure outputs that can be programmed to close at pre-determined times using the playlist.
2. Using the GPIO test fixture, verify that playback of a playlist on the Test Subject can be started by an external contact closure.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.3.4, 7.4.1.6, 7.4.1.7, 7.5.7.2

Test Equipment
DCI Projector
GPIO Test Fixture

Test Material
DCI 2K StEM Test Sequence

## 8.2.6. Interrupt Free Playback

### Objective

Verify that the system can play a sequence of CPLs (a playlist) without noticeable interruptions such as unexpected pauses or visual or audible artifacts.

### Procedures

To verify that playback is possible without any interruptions:

1. Load the DCP *DCI 2K StEM Test Sequence (Encrypted)*, then assemble and start a show.
2. Verify that playback succeeds and is completed without any image or sound distortions and without any interruption.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.4.1.8

Test Material
DCI 2K StEM Test Sequence (Encrypted)

## 8.2.7. Artifact Free Playback of Image Content

### Objective

Verify playback functionality with regard to artifact free playback.

### Procedures

To verify that mode transitions do not cause any artifacts:

1. Assemble a Show Playlist from the DCPs in *DCI 2K Image with Frame Number Burn In (Flat)*, which contains two reels of 1.85:1 content, and *DCI 2K Image with Frame Number Burn In (Scope)*, which contains two reels of 2.39:1 content.
2. Start playback and observe the projected image. Transitions between reels and DCPs are announced visually by means of burned-in subtitles. Verify that for all transitions, no visible artifacts, e.g., rolling are visible.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.4.1.6

Test Material
DCI 2K Image with Frame Number Burn In (Flat)
DCI 2K Image with Frame Number Burn In (Scope)

## 8.2.8. Restarting Playback

### Objective

Verify that power failures cause the system to enter a stable stop/idle condition and that the system provides the ability to restart playback at a point prior to a power interruption.

### Procedures

1. Load the DCP *DCI 2K Image with Frame Number Burn In*, then assemble and start a show.
2. Interrupt the presentation by interrupting the SM's power supply. If possible, the projector power supply should not be interrupted as this may cause overheating and damage the projector.
3. Re-establish power and verify that the system enters a stable stop/idle state. Failure to meet this requirement is cause to fail this test.
4. Verify that the system notifies the user that the last playback was abnormally interrupted, and offers the possibility of restarting the show. Failure to meet this requirement is cause to fail this test.
5. Attempt to restart the presentation at a point prior to the power interruption and verify that the restart was successful. Failure to meet this requirement is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.2.3.13, 7.4.1.2, 7.4.1.8

Test Material
DCI 2K Image with Frame Number Burn In

## 8.2.9. SMS User Accounts

### Objective

Verify that the SMS supports multiple levels of user accounts.

### Procedures

1. Study the user manual to discover factory-created account names and passwords. Failure of the system to provide privilege separation using distinct user accounts is cause to fail this test.
2. If required by the system, create the necessary operating accounts.
3. Return the system to the "logged out" state.
4. For each account, log on to the system using the account information and note the privileges available to the account user (e.g., run show, load content, create account, etc.).
5. Verify that the system can separate, by role, at least the following activities: *a*) create user accounts; *b*) manage content (ingest, delete); *c*) run an existing show. That is, the system must ensure that holding privilege *c* does not allow *b* or *a*, and holding privilege *b* does not allow *a*. Failure of the system to provide this minimum separation of activities is cause to fail this test.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.4.1.3



## 8.2.10. SMS Operator Identification

### Objective

Verify that the security system requires the SMS and SMS operator to identify itself to the Security Manager.

### Procedures

Verify that the security system requires the SMS and SMS operator to identify itself to the Security Manager. This can be implemented by means of the previously mentioned user accounts, for example.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.1.1

## 8.2.11. SMS Identity and Certificate

### Objective

Verify that the SMS (or TMS) carries a [SMPTE-430-2] compliant digital certificate to identify the SMS entity to the SM. Verify that the SMS certificate indicates only the SMS role unless the SMS is contained within a SPB meeting the protection requirements for any other designated roles.

### Procedures

Use network analysis tools (e.g., Wireshark) to monitor the packets exchanged between SMS and SM. Verify that the certificate sent by the SMS during TLS session setup:

1. Is DCI compliant (see Chapter 2: *Digital Cinema Certificates* on how to check a certificate for validity/DCI compliance)
2. Only indicates the SMS role

Note: As an alternative, if the SMS is constructed based on standard computer hardware and software ("off the shelf" operating system, e.g., a Unix derivate), then this test can possibly be done by analysis of the locally stored data. Note: In case the projection system is not available as a whole, or in case the system is not based on standard hardware and software (e.g., standard OS) or the lower OS levels (file systems) are not readily accessible, the above Objective must be verified in cooperation with the manufacturer's system designers.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.2.5

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 8.2.12. Content Keys and TDL check

### Objective

- Verify that the SMS, working with the security infrastructure, checks that the content keys required for playback are available and valid for scheduled exhibitions and the suite equipment to be used for playback is on the TDL.
- Verify that the SMS does this check for every composition individually.

### Procedures

With the test materials specified below, perform the following procedures:

1. Try to assemble and play a show using *DCI 2K StEM Test Sequence (Encrypted)* without providing a KDM. Failure of the SMS to report an error is cause to fail this test. If playback begins this is cause to fail this test.
2. Try to assemble and play a show using *KDM for 2K StEM* and *KDM with incorrect message digest* in that order. The *KDM with incorrect message digest* is invalid (wrong signature/hash error). Failure of the SMS to report an error is cause to fail this test. If playback begins this is cause to fail this test.
3. Try to assemble and play a show using *KDM that has expired* which contains an expired time window. Failure of the SMS to report an error is cause to fail this test. If playback begins this is cause to fail this test.
4. Try to assemble and play a show using *KDM with future validity period* which contains a time window in the future. Failure of the SMS to report an error is cause to fail this test. If playback begins this is cause to fail this test.
5. Try to assemble and play a show using *KDM with empty TDL*, which is a KDM that does not list any trusted devices in its TDL. Failure of the SMS to report an error is cause to fail this test. If playback begins this is cause to fail this test.

Note: The above procedure does not fully cover the objective, as item 2 does not test for all possibilities for KDMs to be invalid. For full coverage, manufacturer assistance is needed.

### Supporting Materials

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.2

Test Material
KDM for 2K StEM
DCI 2K StEM Test Sequence (Encrypted)
KDM with incorrect message digest
KDM that has expired
KDM with future validity period
KDM with empty TDL

**Page Intentionally Left Blank**

---

# Part II. Design Evaluation Guidelines

Version 1 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1 released May 8 2009

**Page Intentionally Left Blank**

---

---

# Chapter 9. FIPS Requirements for a Type 1 SPB

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.0 released May 8 2009

Type 1 Secure Processing Blocks (SPB) are required by DCI to conform to the U.S. National Institute of Standards and Technology (NIST) document *FIPS 140-2: Security Requirements for Cryptographic Modules*. Testing for compliance with FIPS 140-2 is performed by independent laboratories certified by NIST.

The testing program, known as the Cryptographic Module Validation Program (CMVP), is a joint effort of NIST's Computer Systems Laboratory (CSL) and the Communications Security Establishment (CSE) of the Government of Canada. More information about CMVP can be found on the NIST web site at <http://www.csrc.nist.gov/cryptval>. To be compliant with the DCI System Specification, a Type 1 SPB device must be tested by an accredited laboratory, the resulting documentation must be submitted to NIST/CSE for examination, and a validation certificate must be issued by NIST/CSE. Throughout this document, the term "FIPS 140-2 testing" will refer to this entire process.

FIPS 140-2 testing is very thorough but also very selective. To determine whether Type 1 SPB meets the DCI requirements, the documents prepared for and presented to the FIPS testing lab by the manufacturer must be examined by the Test Operator. This chapter will briefly explain the FIPS testing process and the documentation that is produced. A procedure will be presented that will guide the Test Operator through the task of evaluating a FIPS 140-2 test report and determining the DCI compliance status of the respective Test Subject.

## 9.1. FIPS Testing Procedures

This section will explain the process of obtaining a FIPS 140-2 validation certificate from NIST/CSE. This information is intended to guide the Test Operator in understanding the documentation that will be produced in that process. This information is not exhaustive and is not intended to guide a manufacturer in obtaining a validation certificate. The following sub-sections illustrate the tasks in a typical validation process.

### Accredited Laboratory

FIPS 140-2 testing is performed by an accredited laboratory. This Cryptographic Modules Testing (CMT) laboratory will assist the manufacturer in preparing the required documentation and will test sample devices for conformance to the documentation. The CMT laboratory may help the manufacturer resolve compliance issues in the design, but this help is limited to comments on proposed designs, actual design participation may not occur. The documentation and test reports may be submitted to NIST/CSE by the CMT laboratory or the manufacturer.

NIST makes available the list of accredited CMT laboratories on the agency web site (see <http://www.csrc.nist.gov/cryptval/1401labs.htm>). Any of the laboratories can be used, but some restrictions may apply. For example, a laboratory that is owned by the Test Subject manufacturer or one that contributed to the design of the Test Subject will be disqualified from testing that Test Subject. More information about CMT laboratories and laboratory selection can be found in *Frequently Asked Questions for the Cryptographic Module Validation Program* (<http://www.csrc.nist.gov/cryptval/140-1/CMVPFAQ.pdf>).

#### Note

The FIPS 140-2 validation test report prepared by the CMT laboratory is a proprietary and closely controlled document. The manufacturer must ensure that it has permission to disclose the test report to the DCI Testing Organization.

## Standards and Supporting Documentation

The manufacturer must obtain and understand all of the NIST documentation that is relevant to the FIPS 140-2 testing process. In addition to the documentation about the validation process itself, the manufacturer will also need documentation which addresses the requirements for particular algorithms implemented in the device (see <http://www.csrc.nist.gov/cryptval/cavp.htm>).

## Security Element Documentation

All design elements which are addressed by FIPS 140-2, *e.g.*, cryptographic algorithms and Critical Security Parameters (CSP), must be documented and tested according to CMT procedures. The manufacturer must work with the testing lab to identify all such design features and prepare the required documentation. A checklist summarizing the documentation requirements of the standard is found in FIPS PUB 140-2 Appendix A.

## Design Modification

The CAVP and CMVP validation testing processes may require design modifications to the cryptographic module hardware, software, firmware, or documentation. The CMT laboratory performing the validation testing identifies the compliance issues, but does not design or redesign the cryptographic module with the manufacturer, or for the manufacturer.

### Note

The manufacturer is responsible for implementing a compliant design, and submitting required testing evidence to the CMT laboratory for review and testing.

## Test Subject Instrumentation

Where it is not possible to test a particular subsystem from outside the module (*e.g.*, the seed method for a random number generator), the manufacturer must provide the instrumentation necessary to allow the laboratory to test the subsystem. A simulator may be used, for example, to prove the correct functioning of microcode for an ASIC or FPGA.

Additionally, the manufacturer may be required to develop test jigs to facilitate the error injection process; for example, to simulate tamper events and other hardware failures.

See <http://www.csrc.nist.gov/cryptval/140-1/FIPS1402IG.pdf> for more information.

## Operational Testing

The CMT lab exercises the cryptographic module through all major states, including error states, while monitoring all external ports and interfaces using manufacturer testing tools and equipment. This may require the ability to manipulate program execution and record the contents of memory, thus requiring instrumentation as above.

For FIPS 140-2 Level 3, a minimum of five production grade samples of the cryptographic module will be physically attacked and destroyed by the CMT lab during the validation testing process.

## Report Submission

Upon successful completion of the validation testing (no failed test assertions exist), the CMT laboratory submits a FIPS 140-2 validation report to the CMVP for certification. CMVP personnel examine the submission for correctness, sending any necessary requests for clarification to the CMT laboratory. The submission may be rejected, in which case



the manufacturer and laboratory must work to resolve the issue(s) raised and re-submit the validation report. Once the submission is accepted by CMVP, a certificate is issued for the cryptographic module.

The CMVP maintains a list of all cryptographic modules validated to FIPS 140-2 requirements. This list is published on-line at <http://csrc.nist.gov/cryptval/140-1/1401val.htm>. The CMVP also maintains a list of cryptographic modules currently undergoing FIPS 140-2 testing (a listing on the CMVP pre-validation website does not equate to having a FIPS 140-2 validation). The pre-validation list is at <http://csrc.nist.gov/cryptval/preval.htm> [<http://csrc.nist.gov/cryptval/140-1/preval.htm>].

## Maintenance

Changes to the module design require re-validation. The effort required to validate an updated design may be small if the design changes are minor.

## 9.2. Submitted Materials

The CMT laboratory will review and analyze design materials during the validation testing process. A checklist that summarizes the documentation requirements of the standard is found in FIPS PUB 140-2, Appendix A. The following list shows the documents generally expected to be submitted.

- Master Components List (bill of materials); All items submitted as test evidence to the CMT laboratory (*e.g.*, software, firmware, hardware, source code, documentation, etc.) must be specified on the Master Components List, along with a unique identifier and version
- Production grade samples of the cryptographic module (minimum of five for Level 3)
- Security policy
- Data sheets for hardware components
- Listing of all significant information flows
- Finite state model
- Clearly annotated source code
- Functional specifications
- Block diagrams
- Schematics
- VHDL for custom components
- Software design documentation (such as an API or developers guide)
- Mechanical drawings & assembly drawings (approximately to scale)
- Printed circuit board layout drawings
- Cryptographic Key and Critical Security Parameter documentation
- Delivery and operations procedures
- Cryptographic Officer & User guidance

- Configuration management specification
- Operational testing plan(s), and associated testing equipment
- Rationale for exclusion of any components from the security requirements of FIPS 140-2
- Proof of conformance to FCC Part 15, Subpart B Class A requirements
- CAVP Algorithm validation certificates for all implemented Approved cryptographic algorithms
- Documentation detailing the correspondence of all security rules to the implementation

### 9.3. Test Lab Reports

A FIPS 140-2 validation test report is created by CMT laboratory engineers for submission to CMVP. The report details the documentation received and the test engineer's evaluation of the implementation's fidelity to the documentation and FIPS 140-2 requirements.

The following list of documents is taken verbatim from <http://www.csrc.nist.gov/cryptval/140-1/FIPS1402IG.pdf>

- Non-proprietary Security Policy; reference FIPS 140-2 DTR and IG 14.1 for requirements; the non-proprietary security policy shall not be marked as proprietary or copyright without a statement allowing copying or distribution
- CRYPTIK v5.5 (or higher) reports; the validation report submission must be output from the NIST provided Cryptik tool (Cryptik is a proprietary CMVP tool available only to accredited CMT laboratories)
- Physical Test Report (mandatory at Levels 2-4); the laboratory's physical testing report with photos, drawing, etc. as applicable
- Revalidation change summary (if applicable)
- Section Summaries (optional); briefly describes design methods used to meet FIPS 140-2 requirements.

See also <http://www.csrc.nist.gov/cryptval/140-1/FIPS1402IG.pdf>.

### 9.4. Interpreting FIPS Test Reports

The CMT laboratory assessments contained within a FIPS 140-2 validation test report address each of the applicable "TE" requirements corresponding to the eleven areas specified in the FIPS 140-2 Derived Test Requirements (DTR). These requirements instruct the tester as to what he or she must do in order to test the cryptographic module with respect to the given assertion (which is a statement that must be true for the module to satisfy the requirement of a given area at a given level).

For each applicable FIPS 140-2 "TE", the tester's assessment includes:

- A statement that the tester verified the requirement was satisfied, or that the requirement is not applicable.
- Details on how the tester verified the requirement (e.g. through documentation review, source code analysis, physical attack, operational testing, etc.).
- References to supporting design documentation and other test evidence.
- References to algorithm standards and CAVP validation certificates as applicable.

The DCI Testing Organization must obtain an official copy of the FIPS 140-2 validation test report directly from the CMT lab that performed the testing. The Test Operator must verify that the name of the cryptographic module and version (software,

hardware, firmware) under review are identical to the versions reviewed for the FIPS 140-2 validation certificate, and supporting CAVP algorithm validation certificate(s).

To confirm whether the cryptographic module satisfies the DCI requirements, the Test Operator must review the "TE" assessments (and associated references as needed) that are relevant to corresponding DCI requirements (the specific assessments are located below with the respective DCI requirements. The functionality described must be consistent with the observed implementation.

Version 1.0 ARCHIVED ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## 9.5. DCI Requirements for FIPS Modules

Each of the subsections below describes a DCI requirement that must be proven by examining the FIPS 140-2 validation report. For each requirement, observe the design of the respective system element (with the aid of the Test Subject Representative) and record whether or not the design meets the requirement.

### 9.5.1. SM Operating Environment

Verify that the Security Manager (SM) operating environment is limited to the [FIPS-140-2] "limited operational" environment category; "a static non-modifiable virtual operational environment with no underlying general purpose operating system."

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [FIPS-140-2]	9.4.2.4, 9.5.2.5, 9.5.2.7

### 9.5.2. LE Key Generation

Verify the following:

1. That the Security Manager (SM) supports keying of the Link Encryptor (LE) by generating unpredictable keys and having a controlled usage validity period.
2. That Link Encryptor (LE) keys are 112 bits in length for TDES or 128 bits in length for AES, and that those keys are generated according to the requirements of the [DCI-DCSS-1-1], Section 9.7.6 and [FIPS-140-2] level 3 "cryptographic module specification" specifications (per the requirements of [DCI-DCSS-1-1], Section 9.5.2.5).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [FIPS-140-2]	9.4.3.5, 9.4.4, 9.5.2.5, 9.7.6

### 9.5.3. SPB1 Tamper Responsiveness

Verify the following:

1. That SPBs of type 1 are tamper responsive and meet [FIPS-140-2] level 3 "physical security requirements" specifications (per the requirements of [DCI-DCSS-1-1], Section 9.5.2.5).
2. That the SPB monitors SPB physical security protection integrity 24/7 and in the event of intrusion or other tamper detection, terminates all activity, logs the event, and zeroizes all Critical Security Parameters (CSPs) (see [DCI-DCSS-1-1], Section 9.5.2.6).
3. That if the SPB requires a power source to accomplish tamper detection and response, it must zeroize its CSPs prior to any situation arising where such power source may not be available.
4. That log records are not purged in the event of intrusion or other tamper detection.

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Revised May 8 2009

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.2, 9.4.3.6.2.1, 9.4.3.6.3, 9.5.2.1, 9.5.2.2, 9.5.2.5, 9.5.2.6
[FIPS-140-2]	

### 9.5.4. Security Design Description Requirements

Verify that equipment suppliers define and describe their respective security designs surrounding the use of port 1173 per the requirements of [FIPS-140-2] "Cryptographic Module Ports and Interfaces" and the [DCI-DCSS-1-1], Section 9.5.2.5.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3, 9.5.2.5
[FIPS-140-2]	

### 9.5.5. SPB1 Tamper Resistance

Verify that Secure SPBs of type 1 are tamper resistant and meet [FIPS-140-2] level 3 "physical security requirements" specifications (per the requirements of [DCI-DCSS-1-1], Section 9.5.2.5).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.1, 9.5.2.2, 9.5.2.5
[FIPS-140-2]	

### 9.5.6. SPB1 FIPS Requirements

Verify the following:

1. The device is tamper evident.
2. Any maintenance doors are designed with tamper protections in order to prevent access (penetration) other than as permitted.
3. The device meets and is certified for the requirements of [FIPS 140-2] Level 3 in all areas except those subject to the exceptions or additional notes as specified in the [DCI-DCSS-1-1], section 9.5.2.5.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.1, 9.5.2.5
[FIPS-140-2]	

### 9.5.7. SPB1 Secure Silicon FIPS Requirements

Verify that Secure Processing Block (SPB) type 1 secure silicon:

1. Is of the type designed to resist physical and logical attacks, and ensure that a physical attack destroys Critical Security Parameters (CSPs) prior to exposure.

2. Is of the [FIPS-140-2] Physical Security Single-Chip Cryptographic Module type.
3. Meets [FIPS-140-2] level 3 "physical security requirements" specifications (per the requirements of [DCI-DCSS-1-1], Section 9.5.2.5).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.2, 9.5.2.5
[FIPS-140-2]	

## 9.5.8. Asymmetric Key Generation

Verify that keys are generated as specified in [RFC-3447] and per the requirements of [FIPS-140-2] "Cryptographic Key Management" and the [DCI-DCSS-1-1], Section 9.5.2.5.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.5, 9.7.6
[RFC-3447]	

## 9.5.9. Critical Security Parameter Protection

Verify that the following Critical Security Parameters (CSPs) receive Secure Processing Block (SPB) type 1 protection, whenever they exist outside of their originally encrypted state, in accordance with [FIPS-140-2] and the requirements of [DCI-DCSS-1-1], Section 9.5.2.5:

1. Device Private Keys - RSA private key that devices use to prove their identity and facilitate secure Transport Layer Security (TLS) communications.
2. Content Encryption Keys - Key Delivery Message (KDM) AES keys that protect content.
3. Content Integrity Keys - HMAC-SHA-1 keys that protect the integrity of compressed content (integrity pack check parameters).
4. Control Message Encryption and Integrity Keys - AES, HMAC-SHA-1/SHA-256 keys/parameters that protect the privacy and/or integrity of Composition Play Lists, Track File Lists, Key Delivery Messages and other Extra Theatre Messages (ETMs).
5. Link Encryption Keys - Keys that protect the privacy and integrity of uncompressed content for link encryption.
6. TLS secrets - These are transient keys/parameters used or generated in support of TLS and Auditorium Security Messaging (ASM).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.5, 9.5.2.6
[FIPS-140-2]	

## 9.5.10. SPB 1 Firmware Modifications

Verify the following:

Version 1.10 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

1. The device is designed such that the firmware cannot be modified without the knowledge and permission of the original manufacturer.
2. The device's firmware modification procedure requires a digital certificate per [SMPTE-430-2-2006] that identifies the authority figure responsible for making the firmware change.
3. The device logs firmware change information including timestamp, version and operator identity.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.7

**Page Intentionally Left Blank**



---

# Chapter 10. DCI Requirements Review

Like the previous chapter, this chapter contains procedures for evaluating system design for fidelity to DCI requirements that cannot be tested by direct examination of a finished product. These requirements are different though, because they are not proven by the FIPS 140-2 certification process.

The process of proving these requirements is the same, however. Documentation must be produced and Test Subjects must be instrumented to give the Test Operator all necessary information to evaluate the design. Manufacturers must produce proof in the form of design documentation for each of the Requirements listed in Section 10.4.

## 10.1. Type 1 SPB Documentation

For a Type 1 SPB, it should be possible to validate the requirements in this chapter using the test material produced for the FIPS 140-2 test. It may be necessary for the manufacturer to provide additional information in the case where a requirement is not provable using documentation prepared with only the FIPS test in mind. Manufacturers are encouraged to consider the objectives of this chapter when preparing material for the FIPS test of a Type 1 SPB.

The following documents (repeated from Chapter 9) should be used to show compliance with the requirements presented in this chapter:

- Master Components List (bill of materials); All items submitted as test evidence to the Testing Organization (*e.g.*, software, firmware, hardware, source code, documentation, etc.) must be specified on the Master Components List, along with a unique identifier and version
- One (1) production grade sample of the cryptographic module
- Security policy
- Data sheets for hardware components
- Listing of all significant information flows
- Finite state model
- Clearly annotated source code
- Functional specifications
- Block diagrams
- Schematics
- Software design documentation (such as an API or developers guide)
- Mechanical drawings & assembly drawings (approximately to scale)
- Printed circuit board layout drawings
- Cryptographic Key and Critical Security Parameter documentation
- Delivery and operations procedures
- Cryptographic Officer & User guidance

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

- Configuration management specification
- Operational testing plan(s), and associated testing equipment
- Documentation detailing the correspondence of all security rules to the implementation

## 10.2. Type 2 SPB Documentation

For a Type 2 SPB, it is necessary to produce documentation to validate the requirements in this chapter. Because a Type 2 SPB is not required to undergo FIPS 140-2 testing, this documentation will be produced only for the purpose of this DCI test. The same documents listed in Section 10.1 should be created.

## 10.3. Forensic Mark IP Disclosure

For a Test Subject which implements Forensic Marking (FM), it will be necessary to provide, in addition to the documentation listed above, an intellectual property disclosure statement which describes any claims on intellectual property that the manufacturer intends to make on the FM algorithm.

## 10.4. DCI Requirements for Security Modules

Each of the subsections below describes a DCI requirement that must be proven by examining the manufacturer's documentation. For each requirement, observe the design of the respective system element (with the aid of the Test Subject Representative) and record whether or not the design meets the requirement.

### 10.4.1. Theater System Reliability

- Verify that the complete theater system has a reliability, in terms of Mean Time Between Failure (MTBF), of at least 10,000 hours.
- Verify that any failed or malfunctioning unit or component part can be diagnosed and replaced in less than two hours (excluding the time for order and delivery of the replacement part).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.2.3.1, 7.2.3.2

### 10.4.2. Theater System Storage Security

Verify that image and audio essence on storage devices retains the original AES encryption, if present during ingest. It is required that decrypted plaintext (image or audio) essence is never stored on the storage system.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	7.5.3.8

### 10.4.3. Security Devices Self-Test Capabilities

Verify that (to the extent possible) all security devices are designed with self-test capability to announce failures and take themselves out of service.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.1

### 10.4.4. Security Entity Physical Protection

Verify that all functional Security Entities (SE) (except the SMS) are contained within Type 1 SPBs.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.1.1

### 10.4.5. Secure SMS-SM Communication

Verify that the SMS communicates with the SM under its control in a secure fashion (i.e., under TLS).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.1.1

### 10.4.6. Location of Security Manager

- Verify that there is only one Security Manager (SM) for each auditorium.
- Verify that the SM is contained within the Image Media Block (IMB).
- Verify that no SM functions, as defined in [DCI-DCSS-1-1], Sections 9.4.3.5, 9.6.1 and 9.6.1.2, are implemented outside of the secure environment of the IMB.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.2.4, 9.4.3.5, 9.6.1, 9.6.1.2

### 10.4.7. SM Usage of OS Security Features

Verify that the Security Manager (SM) software runs on a Linux kernel configured with the SELinux security extension. Verify that the design uses all appropriate security features of the operating system.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.2.4

### 10.4.8. SM Secure Communications

Verify that the only security communication with systems (processors) external to the Security Manager's (SM) Secure Processing Block (SPB) is by Transport Layer Security (TLS) over a network interface per [DCI-DCSS-1-1], Section 9.4.5.1.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.2.4, 9.4.5.1

### 10.4.9. Playback Preparation

Verify that the SMS prepares the security system for playout within 30 minutes prior to showtime.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

### 10.4.10. SE Uniqueness Constraint

Verify that for normal operations the SM supports suite playback preparation (authentication followed by keying) such that no more than one of each type of SE is enabled (i.e. one image MD, one audio MD, one LDB, one LD/LE). In content owner-approved special case auditorium situations, (e.g. multiple Projector LDBs or LD/LEs), the SM shall support the authentication and keying of multiple Link Encryption operation per the requirements of [DCSS v1.1 Section 9.4.4.1] Multiple Link Encryption Operation.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5, 9.4.4.1

### 10.4.11. Prevention of Keying of Compromised SPBs

Verify that the SM precludes delivery of keys/content to, or playback on, compromised or improperly operating security equipment.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

### 10.4.12. SPB Authentication

Verify that the Security Manager (SM) performs remote Secure Processing Block (SPB) authentication through Transport Layer Security (TLS) session establishment, and maintain the certificate lists collected.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

### 10.4.13. TLS Session Key Refreshes

Verify that the Security Manager (SM) maintains open Transport Layer Security (TLS) sessions for not more than 24 hours between complete restarts (i.e., forces periodic fresh TLS keys).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

### 10.4.14. LE Key Issuance

Verify that the SM supports keying of the Link Encryptor (LE) by transferring LE keys only to an authenticated and trusted Link Decryptor Block (LDB) and companion SPB ( *i.e.*, the projector). Verify that LE keys are not issued to the LDB unless the LDM and companion SPB certificates are both listed on the TDL of the enabling KDM.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

### 10.4.15. Maximum Key Validity Period

Note: Only applicable where external MDs or FM's are used.

Verify that the key usage validity period is less than six hours. Verify that the six hour period does not extend beyond the playback time window specified in the KDM. An exception to this requirement may be made when playout is started within the KDM playout time window, but the playout time window expires before the end of playout. In this case the show may playout beyond the playout time window by a maximum of six (6) hours.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

### 10.4.16. KDM Purge upon Expiry

Verify that the Security Manager (SM) deletes from its storage any Key Delivery Message (KDM) (and associated keys) for which the playout time window has expired (passed). An exception to this requirement may be made when playout is started within the KDM playout time window, but the playout time window expires before the end of playout. In this case the show may playout beyond the playout time window by a maximum of six (6) hours.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

### 10.4.17. Key Usage Time Window

Verify that the Security Manager (SM) enforces the playback time window specified in the Key Delivery Message (KDM) by delivering content keys to Media Decryptors (MD) along with usage periods fully contained within the KDM validity time window. An exception to this requirement may be made when playout is started within the KDM playout time window, but the playout time window expires before the end of playout. In this case the show may playout beyond the playout time window by a maximum of six (6) hours.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.5

### 10.4.18. Projector Secure Silicon Device

Verify that the projector SPB includes a secure silicon host device (see Section 9.5.2 of the DCI Specification v1.1) to support an identity key pair (private key) and appropriate intelligence to support authentication (per implementation options in Section 9.4.3.6.5 of the DCI Specification v1.1), electronic marriage, SPB "open" signal and secure silicon device operational status (e.g., keys zeroed, etc.).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.1

### 10.4.19. Access to Projector Image Signals

Verify that the Projector SPB design does not allow physical access to signals running between the companion SPB and the projector SPB without breaking the marriage.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.1

### 10.4.20. Systems with Electronic Marriage

Verify that an electronic marriage is planned upon installation of an IMB or LDB projector pair. Verify that this physical/electrical connection is battery-backed and monitored 24/7 by the companion SPB and, if broken, shall require a reinstallation (re-marriage) process.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.1

### 10.4.21. Systems Without Electronic Marriage

Verify that in the configuration of a permanently married companion SPB (IMB or LDB) the companion SPB is not field replaceable and require the projector SPB and companion SPB system to both be replaced in the event of an SPB failure.

Verify that the system contains a single digital cinema certificate.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.6.6

### 10.4.22. Clock Date-Time-Range

Verify that the IMB SM clock has a Date-Time range of at least 20 years.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.7

### 10.4.23. Clock Setup

Verify that the IMB SM clock is set by the SM manufacturer to within one second of UTC by means of a national time standard (such as WWV).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.7

### 10.4.24. Clock Stability

Verify that the IMB SM clock is accurate to +/- 30 seconds per month.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.3.7

### 10.4.25. Repair and Renewal of SPBs

Verify that an SPB cannot be repaired or altered without direct manufacturer action.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.3

### 10.4.26. SPB2 Protected Devices

Verify that Type 2 SPB surrounds the following sub-systems:

- a. a security environment consisting of a secure silicon chip; input/output signals to the secure silicon chip and the projector SPB; perimeter access panel monitoring
- b. the projector image signal processing environment

Verify through physical inspection that a sample device contains the above listed sub-systems in a manner consistent with the documentation.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.4

### 10.4.27. Clock Continuity

Verify that the clock is tamper-proof and thereafter may not be reset.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.4.7



### 10.4.28. TLS Endpoints

Verify that all TLS end points are within the physical protection perimeter of the associated SPB.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.1

### 10.4.29. Implementation of RRP

Verify that the RRP approach as described in the DCI Specification v1.1 is followed.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.1

### 10.4.30. SMS and SPB Authentication and ITM Transport Layer

Verify that the authentication of the SPBs and the SMS towards each other utilizes digital certificates as defined by [SMPTE-430-2], which facilitate a cryptographic process that identifies each SPB device to the SM. Verify that the SM and SMS both conduct their intra-theater security messaging under TLS protection (see [RFC-2246]).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.1
[RFC-2246]	

### 10.4.31. Idempotency of ITM RRP

Verify that transactions are "idempotent" (such a transaction may be repeated without changing its outcome).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.1

### 10.4.32. RRP Synchronism

Verify that RRP protocols are synchronous: each pairing must opened and closed before a new RRP is opened between any two devices. Nested transactions (in which one end point must communicate with another end point while the first waits) are allowed.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 2009

### 10.4.33. TLS Mode Bypass Prohibition

Verify that except where noted in the DCI Specification v1.1, non-TLS security communications are not used, and that production Digital Cinema security equipment has no provisions for performing security functions in a TLS "bypass" mode.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3

### 10.4.34. RRP Broadcast Prohibition

Verify that no broadcast RRP commands are used or required.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3

### 10.4.35. Implementation of Proprietary ITMs

Verify that any proprietary ITM implemented by equipment suppliers do not communicate over TCP or UDP port 1173, and that such ITMs do not communicate information that is the subject of any [SMPTE-430-6] commands.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3
[SMPTE-430-6]	

### 10.4.36. RRP Initiator

Verify that, except where noted, only the SMS or SM initiate RRP.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3

### 10.4.37. SPB TLS Session Partners

Verify that SPBs do not communicate with SMs other than the one assigned to their suite.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3

### 10.4.38. SM TLS Session Partners

Verify that SMs do not communicate with SPBs other than those in its suite.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3

### 10.4.39. RRP "Busy" and unsupported Types

Verify that unless otherwise noted, an RRP response is allowed to be busy or an unsupported message type and that such a response is not an error event.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3

### 10.4.40. RRP Operational Message Ports

Verify that Category 1 messages (see Table 15 p.107 of the DCI Specification v1.1) are not transported over TCP port 1173, but another port using TLS.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.5.2.3

### 10.4.41. FM Generic Inserter Requirements

Verify that any pre-processing is able to utilize a single, industry standardized metadata transport format and a generic inserter solution that supports the use of a single image or audio FM technology that generates one set of metadata and uses metadata compatible with all deployed compliant generic inserters. Verify that for the initial generic inserter deployment, the generic inserter in final product form has been openly demonstrated and independently tested to demonstrate compatibility with a minimum of three independent metadata-based forensic marking solutions. Verify that after initial deployment, any subsequent metadata-based FM solutions or generic inserters will function correctly with all deployed compliant systems. Verify that the Forensic Mark processing that generates and inserts markings are real time or faster and occur in a single pass

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.1.1

## 10.4.42. FM Algorithm General Requirements

For a Forensic Marking (FM) embedder:

1. Verify that single distribution inventory is supported by the FM algorithm.
2. Verify by examination of the FM embedder intellectual property disclosure that the terms and conditions of use for the FM algorithm are reasonable and non-discriminatory (RAND).
3. Verify that detection can be performed by the manufacturer or the Rights Owner at the Rights Owner's premises.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.1.1

## 10.4.43. FM Insertion Requirements

Verify that FM insertion is a real-time (i.e., show playback time), in-line process performed in the associated MB, and has a reasonable computational process.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.1.1

## 10.4.44. IFM Visual Transparency

Verify that IFM is visually transparent to the critical viewer in butterfly tests for motion image content.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.1.2

## 10.4.45. IFM Robustness

Verify that IFM resists/survives video processing attacks (such as digital-to-analog conversions, including multiple D-A/A-D conversions), re-sampling and re-quantization (including dithering and recompression), common signal enhancements to image contrast and color, resizing, letterboxing, aperture control, low-pass filtering, anti-aliasing, brick wall filtering, digital video noise reduction filtering, frame-swapping, compression, arbitrary scaling (aspect ratio is not necessarily constant), cropping, overwriting, addition of noise and other transformations. Verify that IFM survives collusion (the combining of multiple videos in the attempt to make a different fingerprint or to remove it), format conversion, the changing of frequencies and spatial resolution (among, for example, NTSC, PAL and SECAM, into another and vice versa), horizontal and vertical shifting and camcorder capture and low bit rate compression (e.g., 500 Kbps H264, 1.1 Mbps MPEG-1).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.1.2

### 10.4.46. AFM Inaudibility

Verify that AFM is inaudible in critical listening A/B tests.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.1.3

### 10.4.47. AFM Robustness

Verify that AFM resists/survives multiple D/A and A/D conversions, radio frequency or infrared transmissions within the theater, any combination and down conversion of captured channels, re-sampling of channels, time compression/expansion with pitch shift and pitch preserved, linear speed changes within 10% and pitch-invariant time scaling within 4%. Verify that AFM resists/survives data reduction coding, nonlinear amplitude compression, additive or multiplicative noise frequency response distortion such as equalization, addition of echo, band-pass filtering, flutter and wow and overdubbing.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.1.3

### 10.4.48. FM Control Instance

Verify that the SM is solely responsible for control of FM marking processes (i.e., "on/off") for the auditorium it is installed in and command and control of this function is only via the KDM indicator per [SMPTE-430-1].

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1] [SMPTE-430-1]	9.4.6.2

### 10.4.49. SE Time Stamping

Verify that SEs or their proxy time stamp log records, with date/time synchronized with the auditorium SM's secure clock.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.1

### 10.4.50. SE Log Authoring

Verify that an SE authors its own log records, or utilizes the services of a proxy within the same secure SPB.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.1

### 10.4.51. SPB Log Storage Requirements

Verify that log records stored in SPBs are stored in non-volatile memory and are not purge-able. Verify that data is overwritten beginning with the oldest data as new log data is accumulated. Verify that no log records are overwritten unless collected by the SM.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.1

### 10.4.52. Remote SPB Log Storage Requirements

Verify that remote SPBs have sufficient secure storage to hold log data to accommodate at least two days worth of typical operation.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.1

### 10.4.53. IMB Log Storage Capabilities

Verify that the IMB SM is capable of storing at least 12 months of typical log data accumulation for the auditorium in which it is installed, including log data collected from the associated remote SPBs.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.1

### 10.4.54. Logging for Standalone Systems

Verify that the logging subsystem implementations do not affect the ability of Exhibition to operate their projection systems in a standalone fashion.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.1

### 10.4.55. Logging of Failed Procedures

Verify that failure or refusal of logged events is also a logged event (as applicable).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.7

### 10.4.56. SPB Log Failure

Verify that behavior of security devices (SPB or SE) is specified and designed to immediately terminate operation, and requires replacement, upon any failure of its secure logging operation.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.10

### 10.4.57. Log Purging in failed SPBs

Verify that resident log records in failed SPBs (and their contained SEs) are not purge-able except by authorized repair centers, which are capable of securely recovering such log records.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.3.10

### 10.4.58. IMB Tasks

Verify that the IMB performs Media Decryption for image essence, performs Forensic Marking for image essence. Verify that after image decryption and FM (and other non-security plain text functions as appropriate by design), the image signal is passed to the projector SPB or LDB, as appropriate. Verify that, if included as part of the IMB SPB design, streaming media decryption and streaming forensic marking for audio is performed and that the audio essence is passed to external components. Verify that, if not part of the IMB SPB, support functions for the remote AMB SPB are performed.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.4.6.6.3

### 10.4.59. Private Keys outside Secure Silicon

Verify that device private keys, whether encrypted or not, do not exist outside of the secure silicon device.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.2

### 10.4.60. Image Keys Outside Secure Silicon

In case keys decrypted from the KDM have to be stored externally (off-secure-chip memory is used for key caching within a Media Decryptor), verify that 2048 bit RSA, 128 bit AES or 112 bit TDES are used for encryption.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.2

### 10.4.61. Prohibition of SPB1 Field Serviceability

Verify that SPBs of Type 1 are not field serviceable (e.g., SPB type 1 maintenance access doors shall not be open-able in the field).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.3

### 10.4.62. Use of Software Protection Methods

Verify that software protection methods are not used to protect CSPs or content essence.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.3

### 10.4.63. TMS role

Verify that in the event that Exhibition command and control designs include the TMS as a device that interfaces with the SMS, such a TMS is viewed by the security system as an SMS, and carries a digital certificate and follows all other SMS behavior, TLS and ITM communications requirements.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.5

### 10.4.64. D-Cinema Security Parameter Protection

Verify that the following Digital Cinema Security Parameters (DCSPs) receive SPB type 1 protection, whenever they exist outside of their originally encrypted state:

1. Watermarking or Fingerprinting command and control - Any of the parameters or keys used in a particular Forensic Marking process.
2. Logged Data - All log event data and associated parameters constituting a log record or report.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.5.2.6

### 10.4.65. RSA Key Entropy

Verify that the mechanism used to generate RSA key pairs must have at least 128-bits of entropy (unpredictability).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.7.6



## 10.4.66. Preloaded Symmetric Key Entropy

Verify that AES or TDES symmetric keys pre-loaded into a device are generated with a high quality random number generator with at least 128 bits of entropy (112 bits for TDES).

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.7.6

## 10.4.67. MD Caching of Keys

Verify that the Media Decryptor is capable of securely caching at least 512 keys.

Reference Document ID	Reference Document Section(s)
[DCI-DCSS-1-1]	9.7.7

**Page Intentionally Left Blank**

---

# Part III. Consolidated Test Procedures

The chapters in this part contain consolidated procedures and standardized test reports for testing Digital Cinema equipment and content. These consolidated procedures refer to the elemental procedures in Part I, building on those procedures to present a complete, ordered sequence for testing the subject.

Version 1.0 ARCHIVED ONLY  
NEWER VERSION AVAILABLE  
Version 1.0 released May 8 2009

**Page Intentionally Left Blank**

---

# Chapter 11. Testing Overview

## 11.1. Test Reports

To prepare a test report, select the report for the test subject and perform the tests in the order presented, recording the results of each test as you progress. When all tests have been performed, count the number of check marks in the greyed-out boxes and record this number in the test session detail as shown in Table 11.1. Complete the other fields in the test session detail and then sign and date the report.

Information about the testing session itself is recorded in the following table for any test sequence performed. All fields must be filled in.

**Table 11.1. Test Session Data**

Reporting date	
Name of Testing Organization	
Address of Testing Organization	1: 2: 3: 4:
Name of Test Operator	
Test location (if not at testing org's site)	1: 2: 3: 4:
Name of Test Subject Representative	
Address of Test Subject Representative	1: 2: 3: 4:
Make and model of Test Subject	
Serial number(s) of Test Subject. Record any applicable values.	
Software version number(s). Record any applicable values.	
Firmware version number(s). Record any applicable values.	

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

Test procedure performed (select one)	DCP Test Standalone Server Test Standalone Projector Test Projector w/Media Block Test
Test status (select one)	Pass Fail

# Chapter 12. Digital Cinema Package (DCP) Consolidated Test Sequence

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.0 released May 8 2009

## 12.1. Overview

This chapter presents a complete sequence of procedures for validating the contents of a Digital Cinema Package (DCP). The tests are drawn primarily from Section 4.6.1. If the DCP contains Composition Playlists with digital signatures, the procedures from Section 2.1 will also be used.

These tests assume that the DCP under test is recorded on a random-access hard disk drive or optical disc using a well-known filesystem (a convention for arranging data on the device). While there are voluntary restrictions on the physical interfaces and filesystems supported for these types of devices, this test may be performed on any volume that can be read by the computer being used to perform the test.

Please note that while this procedure may confirm the correctness of the DCP on the media volume, it does not provide assurance that the media volume itself can be used by the intended playback device. Please consult the manufacturer's documentation to learn which media and filesystem combinations are supported by the intended playout device(s).

## 12.2. DCP Test Sequence

For each of the tables below, follow the instructions in the Procedure column, referring to the appropriate test procedure where referenced. Indicate the status of the test in the Yes, No, and Measured Data columns as instructed. Any marks in greyed-out fields indicate a test failure.

**Table 12.1. Asset Map Procedures**

Step	Procedure	Yes	No	Measured Data
1.	Does the filesystem root contain the filename <code>ASSETMAP.xml</code> ?			
2.	Is the <code>ASSETMAP.xml</code> file a valid SMPTE 429-9 Asset Map per Section 4.1.1? Record the namespace name in the Measured Data field.			
3.	Does the filesystem root contain the filename <code>VOLINDEX.xml</code> ?			
4.	Is the <code>VOLINDEX.xml</code> file a valid SMPTE 429-9 Volume Index per Section 4.1.2? Record the namespace name in the Measured Data field.			
5.	Does the <code>ASSETMAP.xml</code> file reference at least one Packing List file? Record the number of Packing List files referenced.			
6.	For each Chunk element in the <code>ASSETMAP.xml</code> file, check that the filesystem path given in the <code>Path</code> element exists in the filesystem. Record any paths that do not exist.			

Repeat the Packing List sequence for each Packing List in the DCP.

**Table 12.2. Packing List Procedures**

Step	Procedure	Yes	No	Measured Data
7.	Record the filename of the Packing List in the Measured Data field.		None	
8.	Is the Packing List a valid XML structure per Section 4.2.1?			
9.	If the Packing List is signed, is the signature valid per Section 4.2.1?			
10.	Does each Asset element in the Packing List exist in the ASSETMAP.xml file?			
11.	Does the Size in each Asset element match the size of the respective asset?			
12.	Does the Hash in each Asset element match the message digest of the respective asset?			

Repeat the Composition Playlist procedure for each Composition Playlist in the DCP.

**Table 12.3. Composition Playlist Procedures**

Step	Procedure	Yes	No	Measured Data
13.	Record the filename of the Composition Playlist in the Measured Data field.		None	
14.	Is the Composition Playlist a valid XML structure per Section 4.3.1?			
15.	If the Composition Playlist is signed, is the signature valid per Section 4.3.1?			
16.	Does each Asset element in the Composition Playlist reference an asset in the ASSETMAP.xml file? If not, record the UUID values of the missing assets in the Measured Data field.		None	
17.	Does the Hash element in each Asset element (where present) match the value of Hash in the respective Packing List?			

Repeat the Track File procedure for each Track File in the DCP.

**Table 12.4. Track File Procedures**

Step	Procedure	Yes	No	Measured Data
18.	Is the track file an OP-Atom MXF file per the procedure in Section 4.4.2?			
19.	Is the track file at least one second in duration per the procedure in Section 4.4.4? Record the length in the Measured Data field.			
20.	Perform the set of procedures for the appropriate essence type. If the track file contains image essence, perform the procedures in Table 12.5: Image Essence Procedures. If the track file contains sound essence,			



Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
	perform the procedures in Table 12.6: Sound Essence Procedures. Check Yes in this row if the procedure in the essence test sequence succeeds, otherwise check No.			

**Table 12.5. Image Essence Procedures**

Step	Procedure	Yes	No	Measured Data
21.	Are the image parameters in the MXF header correct per the procedure in Section 4.5.1?			
22.	Are the images in the MXF file correctly encoded using JPEG 2000 per the procedure in Section 4.5.1?			

**Table 12.6. Sound Essence Procedures**

Step	Procedure	Yes	No	Measured Data
23.	Does each frame of the track file contain a complete, identically sized set of audio samples (as determined by the sample rate, sample size and channel count metadata) per Section 4.4.6? Record the payload size of the first frame in the Measured Data field.			
24.	Are the audio encoding parameters correct per the procedure in Section 4.5.3?			

**Page Intentionally Left Blank**

# Chapter 13. Digital Cinema Server Consolidated Test Sequence

## 13.1. Overview

The test sequence defined in this chapter is intended to be used to test a stand-alone d-cinema server. The configuration and architecture of the server will vary, but the test sequence assumes that the system consists of at least an Image Media Block (IMB, containing a Security Manager, Media Decryptor, etc.) and a Screen Management Server (SMS). For the purpose of this test, the Test Operator may substitute a Theater Management Server (TMS) for the SMS if it implements the required functionality. Wherever a test procedure refers to an SMS, the equivalent TMS may also be used.

Before performing the test sequence provided below, the Test Operator should read and understand the documentation provided with the Test Subject. If adequate documentation is not available, a Test Subject Representative should be available to provide assistance during the test session.

## 13.2. Server Test Sequence

For each of the tables below, follow the instructions in the Procedure column, referring to the appropriate test procedure where referenced. Indicate the status of the test in the Yes, No, and Measured Data columns as instructed. Any marks in greyed-out fields indicate a test failure. The Test Operator may record any additional observations in the Measured Data Field or on a separate list of notes.

The certificates required by the following three procedures are to be obtained directly from the manufacturer using a trusted channel ( e.g., on a USB memory device received in-person). These certificates will be compared later to those obtained electronically from the Test Subject.

**Table 13.1. Security Manager Certificate**

Step	Procedure	Yes	No	Measured Data
1.	Obtain the X.509 digital certificate associated with the Security Manager and the complete chain of signer certificates up to and including the manufacturer's self-signed root certificate. Validate each certificate using the procedures Section 2.1.1: Basic Certificate Structure through Section 2.1.16: Signature Validation. Check No in this row if any procedure fails on any certificate, otherwise check Yes.			
2.	Using the certificates obtained in the previous step, validate the chain using the procedure in Section 2.1.17: Certificate Chains. Check Yes in this row if the procedure succeeds, otherwise check No.			
3.	Perform the procedure given in Section 5.1.1: SPB Digital Certificate. Record the serial number of the Test Subject in the Measured Data field. Check Yes in this row if the procedure succeeds, otherwise check No.			

The certificates required by the following three procedures are to be obtained directly from the manufacturer using a trusted channel ( e.g., on a USB memory device received in-person). These certificates will be compared later to those obtained electronically from the Test Subject.

**Table 13.2. Screen Manager Certificate**

Step	Procedure	Yes	No	Measured Data
4.	Obtain the X.509 digital certificate associated with the SMS and the complete chain of signer certificates up to and including the manufacturer's self-signed root certificate. Validate each certificate using the procedures Section 2.1.1: Basic Certificate Structure through Section 2.1.16: Signature Validation. Check No in this row if any procedure fails on any certificate, otherwise check Yes.			
5.	Using the certificates obtained in the previous step, validate the chain using the procedure in Section 2.1.17: Certificate Chains. Check Yes in this row if the procedure succeeds, otherwise check No.			
6.	Perform the procedure given in Section 5.1.1: SPB Digital Certificate. Record the serial number of the Test Subject in the Measured Data field. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 13.3. Power**

Step	Procedure	Yes	No	Measured Data
7.	Record the published operating voltage of each power inlet in the Measured Data field. Connect power to the Test Subject as directed by the operating instructions. If the Test Subject does not automatically start when power is applied, follow the manufacturer's power-up instructions to start the system.		None	

**Table 13.4. Operator Roles**

Step	Procedure	Yes	No	Measured Data
8.	Perform the procedure given in Section 8.2.9: SMS User Accounts. Record the available operator roles (names) and whether locally-defined accounts can be created. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 13.5. Screen Management System**

Step	Procedure	Yes	No	Measured Data
9.	Perform the procedure given in Section 8.2.10: SMS Operator Identification. Check Yes in this row if the procedure succeeds, otherwise check No.			
10.	Perform the procedure given in Section 8.2.11: SMS Identity and Certificate. Check Yes in this row if the procedure succeeds, otherwise check No.			

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
11.	Perform the procedure given in Section 8.1.1: Storage System Ingest Interface. Check Yes in this row if the procedure succeeds, otherwise check No.			
12.	Perform the procedure given in Section 8.1.2: Storage System Capacity. Check Yes in this row if the procedure succeeds, otherwise check No.			
13.	Perform the procedure given in Section 8.1.3: Storage System Redundancy. Check Yes in this row if the procedure succeeds, otherwise check No.			
14.	Perform the procedure given in Section 8.1.4: Storage System Performance. Check Yes in this row if the procedure succeeds, otherwise check No.			
15.	Perform the procedure given in Section 8.2: Screen Management System. Check Yes in this row if the procedure succeeds, otherwise check No.			
16.	Perform the procedure given in Section 8.2.2: Show Playlist Creation. Check Yes in this row if the procedure succeeds, otherwise check No.			
17.	Perform the procedure given in Section 8.2.3: Show Playlist Format. Check Yes in this row if the procedure succeeds, otherwise check No.			
18.	Perform the procedure given in Section 8.2.4: KDM Validity Checks. Check Yes in this row if the procedure succeeds, otherwise check No.			
19.	Perform the procedure given in Section 8.2.5: Automation Control and Interfaces. Check Yes in this row if the procedure succeeds, otherwise check No.			
20.	Perform the procedure given in Section 8.2.6: Interrupt Free Playback. Check Yes in this row if the procedure succeeds, otherwise check No.			
21.	Perform the procedure given in Section 8.2.7: Artifact Free Playback of Image Content. Check Yes in this row if the procedure succeeds, otherwise check No.			
22.	Perform the procedure given in Section 8.2.8: Restarting Playback. Check Yes in this row if the procedure succeeds, otherwise check No.			
23.	Perform the procedure given in Section 8.2.12: Content Keys and TDL check. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 13.6. KDM Ingest**

Step	Procedure	Yes	No	Measured Data
24.	Perform the procedure given in Section 3.5.1: KDM NonCriticalExtensions Element. Check Yes in this row if the procedure succeeds, otherwise check No.			

Step	Procedure	Yes	No	Measured Data
25.	Perform the procedure given in Section 3.5.2: ETM IssueDate Field Check. Check Yes in this row if the procedure succeeds, otherwise check No.			
26.	Perform the procedure given in Section 3.5.3: Maximum Number of DCP Keys. Check Yes in this row if the procedure succeeds, otherwise check No.			
27.	Perform the procedure given in Section 3.5.4: Structure ID Check. Check Yes in this row if the procedure succeeds, otherwise check No.			
28.	Perform the procedure given in Section 3.5.5: Certificate Thumbprint Check. Check Yes in this row if the procedure succeeds, otherwise check No.			
29.	Perform the procedure given in Section 3.5.6: Certificate Presence Check. Check Yes in this row if the procedure succeeds, otherwise check No.			
30.	Perform the procedure given in Section 3.5.7: KeyInfo Field Check. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 13.7. Interface**

Step	Procedure	Yes	No	Measured Data
31.	Perform the procedure given in Section 6.6.1: Digital Audio Interfaces. Check Yes in this row if the procedure succeeds, otherwise check No.			
32.	Perform the procedure given in Section 5.2.1: TLS Session Initiation. Check Yes in this row if the procedure succeeds, otherwise check No.			
33.	Perform the procedure given in Section 5.2.2.1: Auditorium Security Message Support. Check Yes in this row if the procedure succeeds, otherwise check No.			
34.	Perform the procedure given in Section 5.2.2.2: ASM Failure Behavior. Check Yes in this row if the procedure succeeds, otherwise check No.			
35.	Perform the procedure given in Section 5.2.2.3: ASM "RRP Invalid". Check Yes in this row if the procedure succeeds, otherwise check No.			
36.	Perform the procedure given in Section 5.2.2.4: ASM "GetTime". Check Yes in this row if the procedure succeeds, otherwise check No.			
37.	Perform the procedure given in Section 5.2.2.5: ASM "GetEventList". Check Yes in this row if the procedure succeeds, otherwise check No.			
38.	Perform the procedure given in Section 5.2.2.6: ASM "GetEventID". Check Yes in this row if the procedure succeeds, otherwise check No.			

Step	Procedure	Yes	No	Measured Data
39.	Perform the procedure given in Section 5.2.2.7: ASM "LEKeyLoad". Check Yes in this row if the procedure succeeds, otherwise check No.			
40.	Perform the procedure given in Section 5.2.2.8: ASM "LEKeyQueryID". Check Yes in this row if the procedure succeeds, otherwise check No.			
41.	Perform the procedure given in Section 5.2.2.9: ASM "LEKeyQueryAll". Check Yes in this row if the procedure succeeds, otherwise check No.			
42.	Perform the procedure given in Section 5.2.2.10: ASM "LEKeyPurgeID". Check Yes in this row if the procedure succeeds, otherwise check No.			
43.	Perform the procedure given in Section 5.2.2.11: ASM "LEKeyPurgeAll". Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 13.8. Log Reporting**

Step	Procedure	Yes	No	Measured Data
44.	Perform the procedure given in Section 5.3.2.1: Log Structure. Check Yes in this row if the procedure succeeds, otherwise check No.			
45.	Perform the procedure given in Section 5.3.2.2: Log Records for Multiple SPBs. Check Yes in this row if the procedure succeeds, otherwise check No.			
46.	Perform the procedure given in Section 5.3.2.3: Log Sequence Numbers. Check Yes in this row if the procedure succeeds, otherwise check No.			
47.	Perform the procedure given in Section 5.3.2.4: Log Collection by the SM. Check Yes in this row if the procedure succeeds, otherwise check No.			
48.	Perform the procedure given in Section 5.3.2.5: General Log System Failure. Check Yes in this row if the procedure succeeds, otherwise check No.			
49.	Perform the procedure given in Section 5.3.3.1: SM Proxy of Log Events. Check Yes in this row if the procedure succeeds, otherwise check No.			
50.	Perform the procedure given in Section 5.3.3.2: SM Proxy of Security Operations Events. Check Yes in this row if the procedure succeeds, otherwise check No.			
51.	Perform the procedure given in Section 5.3.3.3: SM Proxy of Security ASM Events. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 13.9. Security Events**

Step	Procedure	Yes	No	Measured Data
52.	Perform the procedure given in Section 5.4.1.1: FrameSequencePlayed Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
53.	Perform the procedure given in Section 5.4.1.2: CPLStart Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
54.	Perform the procedure given in Section 5.4.1.3: CPLEnd Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
55.	Perform the procedure given in Section 5.4.1.4: PlayoutComplete Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
56.	Perform the procedure given in Section 5.4.1.5: CPLCheck Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
57.	Perform the procedure given in Section 5.4.1.6: KDMKeysReceived Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
58.	Perform the procedure given in Section 5.4.1.7: KDMDeleted Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
59.	Perform the procedure given in Section 5.4.2.1: LinkOpened Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
60.	Perform the procedure given in Section 5.4.2.2: LinkClosed Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
61.	Perform the procedure given in Section 5.4.2.3: LinkException Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
62.	Perform the procedure given in Section 5.4.2.4: LogTransfer Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
63.	Perform the procedure given in Section 5.4.2.5: KeyTransfer Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
64.	Perform the procedure given in Section 5.4.2.6: SPBStartup SPBShutdown Events. Check Yes in this row if the procedure succeeds, otherwise check No.			
65.	Perform the procedure given in Section 5.4.2.8: SPBClockadjust Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
66.	Perform the procedure given in Section 5.4.2.10: SPBSoftware Event. Check Yes in this row if the procedure succeeds, otherwise check No.			



**Table 13.10. Essence Reproduction**

Step	Procedure	Yes	No	Measured Data
67.	Perform the procedure given in Section 6.5.2: Decoder Requirements. Check Yes in this row if the procedure succeeds, otherwise check No.			
68.	Perform the procedure given in Section 6.5.1: Playback of Image Only Material. Check Yes in this row if the procedure succeeds, otherwise check No.			
69.	Perform the procedure given in Section 6.1.1: Image Integrity Checking. Check Yes in this row if the procedure succeeds, otherwise check No.			
70.	Perform the procedure given in Section 6.6.2: Audio Sample Rate Conversion. Check Yes in this row if the procedure succeeds, otherwise check No.			
71.	Perform the procedure given in Section 6.6.3: Audio Delay Setup. Check Yes in this row if the procedure succeeds, otherwise check No.			
72.	Perform the procedure given in Section 6.6.4: Click Free Splicing of Audio Track Files. Check Yes in this row if the procedure succeeds, otherwise check No.			
73.	Perform the procedure given in Section 6.1.2: Sound Integrity Checking. Check Yes in this row if the procedure succeeds, otherwise check No.			
74.	Perform the procedure given in Section 6.7.1: Media Block Overlay. Check Yes in this row if the procedure succeeds, otherwise check No.			
75.	Perform the procedure given in Section 6.7.2: Timed Text Synchronization. Check Yes in this row if the procedure succeeds, otherwise check No.			
76.	Perform the procedure given in Section 6.7.3: Support for Multiple Captions. Check Yes in this row if the procedure succeeds, otherwise check No.			
77.	Perform the procedure given in Section 6.7.4: Default Timed Text Font. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 13.11. Media Block Security**

Step	Procedure	Yes	No	Measured Data
78.	Perform the procedure given in Section 6.1.3: Restriction of Keying to Monitored Link Decryptors. Check Yes in this row if the procedure succeeds, otherwise check No.			
79.	Perform the procedure given in Section 6.1.4: Restriction of Keying to MD Type. Check Yes in this row if the procedure succeeds, otherwise check No.			
80.	Perform the procedure given in Section 6.1.5: Restriction of Keying to valid CPLs. Check Yes in			

Step	Procedure	Yes	No	Measured Data
	this row if the procedure succeeds, otherwise check No.			
81.	Perform the procedure given in Section 6.1.6: Remote SPB Integrity Monitoring. Check Yes in this row if the procedure succeeds, otherwise check No.			
82.	Perform the procedure given in Section 6.1.7: SPB Integrity Fault Consequences. Check Yes in this row if the procedure succeeds, otherwise check No.			
83.	Perform the procedure given in Section 6.2.1: LDB Trust. Check Yes in this row if the procedure succeeds, otherwise check No.			
84.	Perform the procedure given in Section 6.2.2: Multiple LE Operation. Check Yes in this row if the procedure succeeds, otherwise check No.			
85.	Perform the procedure given in Section 6.2.3: LE Key Usage. Check Yes in this row if the procedure succeeds, otherwise check No.			
86.	Perform the procedure given in Section 6.2.4: IMB Link Encryption. Check Yes in this row if the procedure succeeds, otherwise check No.			
87.	Perform the procedure given in Section 6.3.1: Clock Adjustment. Check Yes in this row if the procedure succeeds, otherwise check No.			
88.	Perform the procedure given in Section 6.3.2: Clock Battery. Check Yes in this row if the procedure succeeds, otherwise check No.			
89.	Perform the procedure given in Section 6.3.3: Clock Resolution. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 13.12. Forensic Marking**

Step	Procedure	Yes	No	Measured Data
90.	Perform the procedure given in Section 6.4.1: FM Application Constraints. Check Yes in this row if the procedure succeeds, otherwise check No.			
91.	Perform the procedure given in Section 6.4.2: Granularity of FM Control. Check Yes in this row if the procedure succeeds, otherwise check No.			
92.	Perform the procedure given in Section 6.4.3: FM Payload. Check Yes in this row if the procedure succeeds, otherwise check No.			

# Chapter 14. Digital Cinema Projector Consolidated Test Sequence

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

## 14.1. Overview

The test sequence defined in this chapter is intended to be used to test a stand-alone d-cinema projector. The configuration and architecture of the projector will vary, but the test sequence assumes that the system consists of at least a Link Decryptor Block (LD) and a light processing system including electronic and optical components.

Before performing the test sequence provided below, the Test Operator should read and understand the documentation provided with the Test Subject. If adequate documentation is not available, a Test Subject Representative should be available to provide assistance during the test session.

## 14.2. Projector Test Sequence

For each of the tables below, follow the instructions in the Procedure column, referring to the appropriate test procedure where referenced. Indicate the status of the test in the Yes, No, and Measured Data columns as instructed. Any marks in greyed-out fields indicate a test failure. The Test Operator may record any additional observations in the Measured Data Field or on a separate list of notes.

The certificates required by the following three procedures are to be obtained directly from the manufacturer using a trusted channel ( e.g., on a USB memory device received in-person). These certificates will be compared later to those obtained electronically from the Test Subject.

**Table 14.1. Projector Certificate**

Step	Procedure	Yes	No	Measured Data
1.	Obtain the X.509 digital certificate associated with the Type 2 SPB and the complete chain of signer certificates up to and including the manufacturer's self-signed root certificate. Validate each certificate using the procedures Section 2.1.1: Basic Certificate Structure through Section 2.1.16: Signature Validation. Check No in this row if any procedure fails on any certificate, otherwise check Yes.			
2.	Using the certificates obtained in the previous step, validate the chain using the procedure in Section 2.1.17: Certificate Chains. Check Yes in this row if the procedure succeeds, otherwise check No.			
3.	Perform the procedure given in Section 5.1.1: SPB Digital Certificate. Record the serial number of the Test Subject in the Measured Data field. Check Yes in this row if the procedure succeeds, otherwise check No.			

The certificates required by the following three procedures are to be obtained directly from the manufacturer using a trusted channel ( e.g., on a USB memory device received in-person). These certificates will be compared later to those obtained electronically from the Test Subject.

**Table 14.2. Link Decryptor Certificate**

Step	Procedure	Yes	No	Measured Data
4.	Obtain the X.509 digital certificate associated with the Link Decryptor and the complete chain of signer certificates up to and including the manufacturer's self-signed root certificate. Validate each certificate using the procedures Section 2.1.1: Basic Certificate Structure through Section 2.1.16: Signature Validation. Check No in this row if any procedure fails on any certificate, otherwise check Yes.			
5.	Using the certificates obtained in the previous step, validate the chain using the procedure in Section 2.1.17: Certificate Chains. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 14.3. Power**

Step	Procedure	Yes	No	Measured Data
6.	Record the published operating voltage of each power inlet in the Measured Data field. Connect power to the Test Subject as directed by the operating instructions. If the Test Subject does not automatically start when power is applied, follow the manufacturer's power-up instructions to start the system.		None	
7.	Perform the procedure given in Section 5.1.1: SPB Digital Certificate. Record the serial number of the Test Subject in the Measured Data field. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 14.4. Secure Processing Block Type 2**

Step	Procedure	Yes	No	Measured Data
8.	Perform the procedure given in Section 5.1.2: SPB Type 2 Security Perimeter. Check Yes in this row if the procedure succeeds, otherwise check No.			
9.	Perform the procedure given in Section 5.1.3: SPB Type 2 Secure Silicon. Check Yes in this row if the procedure succeeds, otherwise check No.			
10.	Perform the procedure given in Section 7.2.3: SPB2 Requirements. Check Yes in this row if the procedure succeeds, otherwise check No.			
11.	Perform the procedure given in Section 7.2.4: SPB2 Secure Silicon Requirements. Check Yes in this row if the procedure succeeds, otherwise check No.			
12.	Perform the procedure given in Section 7.2.5: SPB2 Tamper Evidence. Check Yes in this row if the procedure succeeds, otherwise check No.			

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
13.	Perform the procedure given in Section 7.2.6: SPB2 Secure Silicon Field Replacement. Check Yes in this row if the procedure succeeds, otherwise check No.			
14.	Perform the procedure given in Section 7.2.1: Projector Physical Protection. Check Yes in this row if the procedure succeeds, otherwise check No.			
15.	Perform the procedure given in Section 7.3.1: Projector Companion SPB Location. Check Yes in this row if the procedure succeeds, otherwise check No.			
16.	Perform the procedure given in Section 7.2.7: Systems Without Electronic Marriage. Check Yes in this row if the procedure succeeds, otherwise check No.			
17.	Perform the procedure given in Section 7.2.8: Electronic Marriage Break Key Retaining. Check Yes in this row if the procedure succeeds, otherwise check No.			
18.	Perform the procedure given in Section 7.2.2: Projector Access Door. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 14.5. Interface**

Step	Procedure	Yes	No	Measured Data
19.	Perform the procedure given in Section 5.2.1: TLS Session Initiation. Check Yes in this row if the procedure succeeds, otherwise check No.			
20.	Perform the procedure given in Section 5.2.2.1: Auditorium Security Message Support. Check Yes in this row if the procedure succeeds, otherwise check No.			
21.	Perform the procedure given in Section 5.2.2.2: ASM Failure Behavior. Check Yes in this row if the procedure succeeds, otherwise check No.			
22.	Perform the procedure given in Section 5.2.2.4: ASM "GetTime". Check Yes in this row if the procedure succeeds, otherwise check No.			
23.	Perform the procedure given in Section 5.2.2.5: ASM "GetEventList". Check Yes in this row if the procedure succeeds, otherwise check No.			
24.	Perform the procedure given in Section 5.2.2.6: ASM "GetEventID". Check Yes in this row if the procedure succeeds, otherwise check No.			
25.	Perform the procedure given in Section 5.2.2.7: ASM "LEKeyLoad". Check Yes in this row if the procedure succeeds, otherwise check No.			

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
26.	Perform the procedure given in Section 5.2.2.8: ASM "LEKeyQueryID". Check Yes in this row if the procedure succeeds, otherwise check No.			
27.	Perform the procedure given in Section 5.2.2.9: ASM "LEKeyQueryAll". Check Yes in this row if the procedure succeeds, otherwise check No.			
28.	Perform the procedure given in Section 5.2.2.10: ASM "LEKeyPurgeID". Check Yes in this row if the procedure succeeds, otherwise check No.			
29.	Perform the procedure given in Section 5.2.2.11: ASM "LEKeyPurgeAll". Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 14.6. Security Events**

Step	Procedure	Yes	No	Measured Data
30.	Perform the procedure given in Section 5.4.2.1: LinkOpened Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
31.	Perform the procedure given in Section 5.4.2.2: LinkClosed Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
32.	Perform the procedure given in Section 5.4.2.3: LinkException Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
33.	Perform the procedure given in Section 5.4.2.4: LogTransfer Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
34.	Perform the procedure given in Section 5.4.2.5: KeyTransfer Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
35.	Perform the procedure given in Section 5.4.2.6: SPBStartup SPBShutdown Events. Check Yes in this row if the procedure succeeds, otherwise check No.			
36.	Perform the procedure given in Section 5.4.2.8: SPBClockadjust Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
37.	Perform the procedure given in Section 5.4.2.10: SPBSoftware Event. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 14.7. Link Decryptor**

Step	Procedure	Yes	No	Measured Data
38.	Perform the procedure given in Section 7.4.1: LDB without Electronic Marriage. Check Yes in this row if the procedure succeeds, otherwise check No.			

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
39.	Perform the procedure given in Section 7.3.3: Companion SPB Marriage Break Key Retaining. Check Yes in this row if the procedure succeeds, otherwise check No.			
40.	Perform the procedure given in Section 7.4.2: LDB TLS Session Constraints. Check Yes in this row if the procedure succeeds, otherwise check No.			
41.	Perform the procedure given in Section 7.4.3: LDB Time-Awareness. Check Yes in this row if the procedure succeeds, otherwise check No.			
42.	Perform the procedure given in Section 7.4.4: LDB ASM Conformity. Check Yes in this row if the procedure succeeds, otherwise check No.			
43.	Perform the procedure given in Section 7.4.5: LDB Key Storage. Check Yes in this row if the procedure succeeds, otherwise check No.			
44.	Perform the procedure given in Section 7.4.6: LDB Key Purging. Check Yes in this row if the procedure succeeds, otherwise check No.			
45.	Perform the procedure given in Section 7.4.7: LDB Logging. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 14.8. Image Processing**

Step	Procedure	Yes	No	Measured Data
46.	Perform the procedure given in Section 7.5.1: Projector Overlay. Check Yes in this row if the procedure succeeds, otherwise check No.			
47.	Perform the procedure given in Section 7.5.2: Projector Lens. Check Yes in this row if the procedure succeeds, otherwise check No.			
48.	Perform the procedure given in Section 7.5.3: Projector Pixel Count/Structure. Check Yes in this row if the procedure succeeds, otherwise check No.			
49.	Perform the procedure given in Section 7.5.4: Projector Spatial Resolution and Frame Rate Conversion. Check Yes in this row if the procedure succeeds, otherwise check No.			
50.	Perform the procedure given in Section 7.5.5: White Point Luminance and Uniformity. Check Yes in this row if the procedure succeeds, otherwise check No.			
51.	Perform the procedure given in Section 7.5.6: White Point Chromaticity and Uniformity. Check Yes in this row if the procedure succeeds, otherwise check No.			
52.				

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
	Perform the procedure given in Section 7.5.7: Sequential Contrast. Check Yes in this row if the procedure succeeds, otherwise check No.			
53.	Perform the procedure given in Section 7.5.8: Intra-frame Contrast. Check Yes in this row if the procedure succeeds, otherwise check No.			
54.	Perform the procedure given in Section 7.5.9: Grayscale Tracking. Check Yes in this row if the procedure succeeds, otherwise check No.			
55.	Perform the procedure given in Section 7.5.10: Contouring. Check Yes in this row if the procedure succeeds, otherwise check No.			
56.	Perform the procedure given in Section 7.5.11: Transfer Function. Check Yes in this row if the procedure succeeds, otherwise check No.			
57.	Perform the procedure given in Section 7.5.12: Color Accuracy. Check Yes in this row if the procedure succeeds, otherwise check No.			



# Chapter 15. Digital Cinema Projector with IMB Consolidated Test Sequence

Version 1.1 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

## 15.1. Overview

The test sequence defined in this chapter is intended to be used to test a d-cinema projector with an integrated Image Media Block (IMB). The configuration and architecture of the system will vary, but the test sequence assumes that the system consists of at least a light processing system including electronic and optical components, an Image Media Block (containing a Security Manager, Media Decryptor, etc.), and a Screen Management Server (SMS). For the purpose of this test, the Test Operator may substitute a Theater Management Server (TMS) for the SMS if it implements the required functionality. Wherever a test procedure refers to an SMS, the equivalent TMS may also be used.

Before performing the test sequence provided below, the Test Operator should read and understand the documentation provided with the Test Subject. If adequate documentation is not available, a Test Subject Representative should be available to provide assistance during the test session.

## 15.2. Projector Test Sequence

For each of the tables below, follow the instructions in the Procedure column, referring to the appropriate test procedure where referenced. Indicate the status of the test in the Yes, No, and Measured Data columns as instructed. Any marks in greyed-out fields indicate a test failure. The Test Operator may record any additional observations in the Measured Data Field or on a separate list of notes.

The certificates required by the following four sequence procedures are to be obtained directly from the manufacturer using a trusted channel ( e.g., on a USB memory device received in-person). These certificates will be compared later to those obtained electronically from the Test Subject.

**Table 15.1. Security Manager Certificate**

Step	Procedure	Yes	No	Measured Data
1.	Obtain the X.509 digital certificate associated with the Security Manager and the complete chain of signer certificates up to and including the manufacturer's self-signed root certificate. Validate each certificate using the procedures Section 2.1.1: Basic Certificate Structure through Section 2.1.16: Signature Validation. Check No in this row if any procedure fails on any certificate, otherwise check Yes.			
2.	Using the certificates obtained in the previous step, validate the chain using the procedure in Section 2.1.17: Certificate Chains. Check Yes in this row if the procedure succeeds, otherwise check No.			
3.	Perform the procedure given in Section 5.1.1: SPB Digital Certificate. Record the serial number of the Test Subject in the Measured Data field. Check Yes			

Step	Procedure	Yes	No	Measured Data
	in this row if the procedure succeeds, otherwise check No.			

The certificates required by the following three procedures are to be obtained directly from the manufacturer using a trusted channel ( e.g., on a USB memory device received in-person). These certificates will be compared later to those obtained electronically from the Test Subject.

**Table 15.2. Screen Manager Certificate**

Step	Procedure	Yes	No	Measured Data
4.	Obtain the X.509 digital certificate associated with the SMS and the complete chain of signer certificates up to and including the manufacturer's self-signed root certificate. Validate each certificate using the procedures Section 2.1.1: Basic Certificate Structure through Section 2.1.16: Signature Validation. Check No in this row if any procedure fails on any certificate, otherwise check Yes.			
5.	Using the certificates obtained in the previous step, validate the chain using the procedure in Section 2.1.17: Certificate Chains. Check Yes in this row if the procedure succeeds, otherwise check No.			
6.	Perform the procedure given in Section 5.1.1: SPB Digital Certificate. Record the serial number of the Test Subject in the Measured Data field. Check Yes in this row if the procedure succeeds, otherwise check No.			

The certificates required by the following three procedures are to be obtained directly from the manufacturer using a trusted channel ( e.g., on a USB memory device received in-person). These certificates will be compared later to those obtained electronically from the Test Subject.

**Table 15.3. Projector Certificate**

Step	Procedure	Yes	No	Measured Data
7.	Obtain the X.509 digital certificate associated with the Type 2 SPB and the complete chain of signer certificates up to and including the manufacturer's self-signed root certificate. Validate each certificate using the procedures Section 2.1.1: Basic Certificate Structure through Section 2.1.16: Signature Validation. Check No in this row if any procedure fails on any certificate, otherwise check Yes.			
8.	Using the certificates obtained in the previous step, validate the chain using the procedure in Section 2.1.17: Certificate Chains. Check Yes in this row if the procedure succeeds, otherwise check No.			
9.	Perform the procedure given in Section 5.1.1: SPB Digital Certificate. Record the serial number of the			

Step	Procedure	Yes	No	Measured Data
	Test Subject in the Measured Data field. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.4. Power**

Step	Procedure	Yes	No	Measured Data
10.	Record the published operating voltage of each power inlet in the Measured Data field. Connect power to the Test Subject as directed by the operating instructions. If the Test Subject does not automatically start when power is applied, follow the manufacturer's power-up instructions to start the system.		None	

**Table 15.5. Operator Roles**

Step	Procedure	Yes	No	Measured Data
11.	Perform the procedure given in Section 8.2.9: SMS User Accounts. Record the available operator roles (names) and whether locally-defined accounts can be created. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.6. Screen Management System**

Step	Procedure	Yes	No	Measured Data
12.	Perform the procedure given in Section 8.2.10: SMS Operator Identification. Check Yes in this row if the procedure succeeds, otherwise check No.			
13.	Perform the procedure given in Section 8.2.11: SMS Identity and Certificate. Check Yes in this row if the procedure succeeds, otherwise check No.			
14.	Perform the procedure given in Section 8.1.1: Storage System Ingest Interface. Check Yes in this row if the procedure succeeds, otherwise check No.			
15.	Perform the procedure given in Section 8.1.2: Storage System Capacity. Check Yes in this row if the procedure succeeds, otherwise check No.			
16.	Perform the procedure given in Section 8.1.3: Storage System Redundancy. Check Yes in this row if the procedure succeeds, otherwise check No.			
17.	Perform the procedure given in Section 8.1.4: Storage System Performance. Check Yes in this row if the procedure succeeds, otherwise check No.			
18.	Perform the procedure given in Section 8.2: Screen Management System. Check Yes in this row if the procedure succeeds, otherwise check No.			
19.				

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
	Perform the procedure given in Section 8.2.2: Show Playlist Creation. Check Yes in this row if the procedure succeeds, otherwise check No.			
20.	Perform the procedure given in Section 8.2.3: Show Playlist Format. Check Yes in this row if the procedure succeeds, otherwise check No.			
21.	Perform the procedure given in Section 8.2.4: KDM Validity Checks. Check Yes in this row if the procedure succeeds, otherwise check No.			
22.	Perform the procedure given in Section 8.2.5: Automation Control and Interfaces. Check Yes in this row if the procedure succeeds, otherwise check No.			
23.	Perform the procedure given in Section 8.2.6: Interrupt Free Playback. Check Yes in this row if the procedure succeeds, otherwise check No.			
24.	Perform the procedure given in Section 8.2.7: Artifact Free Playback of Image Content. Check Yes in this row if the procedure succeeds, otherwise check No.			
25.	Perform the procedure given in Section 8.2.8: Restarting Playback. Check Yes in this row if the procedure succeeds, otherwise check No.			
26.	Perform the procedure given in Section 8.2.12: Content Keys and TDL check. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.7. KDM Ingest**

Step	Procedure	Yes	No	Measured Data
27.	Perform the procedure given in Section 3.5.1: KDM NonCriticalExtensions Element. Check Yes in this row if the procedure succeeds, otherwise check No.			
28.	Perform the procedure given in Section 3.5.2: ETM IssueDate Field Check. Check Yes in this row if the procedure succeeds, otherwise check No.			
29.	Perform the procedure given in Section 3.5.3: Maximum Number of DCP Keys. Check Yes in this row if the procedure succeeds, otherwise check No.			
30.	Perform the procedure given in Section 3.5.4: Structure ID Check. Check Yes in this row if the procedure succeeds, otherwise check No.			
31.	Perform the procedure given in Section 3.5.5: Certificate Thumbprint Check. Check Yes in this row if the procedure succeeds, otherwise check No.			
32.	Perform the procedure given in Section 3.5.6: Certificate Presence Check. Check Yes in this row if the procedure succeeds, otherwise check No.			

Step	Procedure	Yes	No	Measured Data
33.	Perform the procedure given in Section 3.5.7: KeyInfo Field Check. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.8. Interface**

Step	Procedure	Yes	No	Measured Data
34.	Perform the procedure given in Section 6.6.1: Digital Audio Interfaces. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.9. Log Reporting**

Step	Procedure	Yes	No	Measured Data
35.	Perform the procedure given in Section 5.3.2.1: Log Structure. Check Yes in this row if the procedure succeeds, otherwise check No.			
36.	Perform the procedure given in Section 5.3.2.2: Log Records for Multiple SPBs. Check Yes in this row if the procedure succeeds, otherwise check No.			
37.	Perform the procedure given in Section 5.3.2.3: Log Sequence Numbers. Check Yes in this row if the procedure succeeds, otherwise check No.			
38.	Perform the procedure given in Section 5.3.2.4: Log Collection by the SM. Check Yes in this row if the procedure succeeds, otherwise check No.			
39.	Perform the procedure given in Section 5.3.2.5: General Log System Failure. Check Yes in this row if the procedure succeeds, otherwise check No.			
40.	Perform the procedure given in Section 5.3.3.1: SM Proxy of Log Events. Check Yes in this row if the procedure succeeds, otherwise check No.			
41.	Perform the procedure given in Section 5.3.3.2: SM Proxy of Security Operations Events. Check Yes in this row if the procedure succeeds, otherwise check No.			
42.	Perform the procedure given in Section 5.3.3.3: SM Proxy of Security ASM Events. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.10. Security Events**

Step	Procedure	Yes	No	Measured Data
43.	Perform the procedure given in Section 5.4.1.1: FrameSequencePlayed Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
44.	Perform the procedure given in Section 5.4.1.2: CPLStart Event. Check Yes in this row if the procedure succeeds, otherwise check No.			

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
45.	Perform the procedure given in Section 5.4.1.3: CPLEnd Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
46.	Perform the procedure given in Section 5.4.1.4: PayoutComplete Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
47.	Perform the procedure given in Section 5.4.1.5: CPLCheck Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
48.	Perform the procedure given in Section 5.4.1.6: KDMKeysReceived Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
49.	Perform the procedure given in Section 5.4.1.7: KDMDeleted Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
50.	Perform the procedure given in Section 5.4.2.6: SPBStartup SPBShutdown Events. Check Yes in this row if the procedure succeeds, otherwise check No.			
51.	Perform the procedure given in Section 5.4.2.8: SPBClockadjust Event. Check Yes in this row if the procedure succeeds, otherwise check No.			
52.	Perform the procedure given in Section 5.4.2.10: SPBSoftware Event. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.11. Essence Reproduction**

Step	Procedure	Yes	No	Measured Data
53.	Perform the procedure given in Section 6.5.2: Decoder Requirements. Check Yes in this row if the procedure succeeds, otherwise check No.			
54.	Perform the procedure given in Section 6.5.1: Playback of Image Only Material. Check Yes in this row if the procedure succeeds, otherwise check No.			
55.	Perform the procedure given in Section 6.1.1: Image Integrity Checking. Check Yes in this row if the procedure succeeds, otherwise check No.			
56.	Perform the procedure given in Section 6.6.2: Audio Sample Rate Conversion. Check Yes in this row if the procedure succeeds, otherwise check No.			
57.	Perform the procedure given in Section 6.6.3: Audio Delay Setup. Check Yes in this row if the procedure succeeds, otherwise check No.			
58.	Perform the procedure given in Section 6.6.4: Click Free Splicing of Audio Track Files. Check Yes in this row if the procedure succeeds, otherwise check No.			

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
59.	Perform the procedure given in Section 6.1.2: Sound Integrity Checking. Check Yes in this row if the procedure succeeds, otherwise check No.			
60.	Perform the procedure given in Section 6.7.1: Media Block Overlay. Check Yes in this row if the procedure succeeds, otherwise check No.			
61.	Perform the procedure given in Section 6.7.2: Timed Text Synchronization. Check Yes in this row if the procedure succeeds, otherwise check No.			
62.	Perform the procedure given in Section 6.7.3: Support for Multiple Captions. Check Yes in this row if the procedure succeeds, otherwise check No.			
63.	Perform the procedure given in Section 6.7.4: Default Timed Text Font. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.12. Media Block Security**

Step	Procedure	Yes	No	Measured Data
64.	Perform the procedure given in Section 6.1.3: Restriction of Keying to Monitored Link Decryptors. Check Yes in this row if the procedure succeeds, otherwise check No.			
65.	Perform the procedure given in Section 6.1.4: Restriction of Keying to MD Type. Check Yes in this row if the procedure succeeds, otherwise check No.			
66.	Perform the procedure given in Section 6.1.5: Restriction of Keying to valid CPLs. Check Yes in this row if the procedure succeeds, otherwise check No.			
67.	Perform the procedure given in Section 6.1.6: Remote SPB Integrity Monitoring. Check Yes in this row if the procedure succeeds, otherwise check No.			
68.	Perform the procedure given in Section 6.1.7: SPB Integrity Fault Consequences. Check Yes in this row if the procedure succeeds, otherwise check No.			
69.	Perform the procedure given in Section 6.2.1: LDB Trust. Check Yes in this row if the procedure succeeds, otherwise check No.			
70.	Perform the procedure given in Section 6.2.2: Multiple LE Operation. Check Yes in this row if the procedure succeeds, otherwise check No.			
71.	Perform the procedure given in Section 6.2.3: LE Key Usage. Check Yes in this row if the procedure succeeds, otherwise check No.			
72.				

Step	Procedure	Yes	No	Measured Data
	Perform the procedure given in Section 6.2.4: IMB Link Encryption. Check Yes in this row if the procedure succeeds, otherwise check No.			
73.	Perform the procedure given in Section 6.3.1: Clock Adjustment. Check Yes in this row if the procedure succeeds, otherwise check No.			
74.	Perform the procedure given in Section 6.3.2: Clock Battery. Check Yes in this row if the procedure succeeds, otherwise check No.			
75.	Perform the procedure given in Section 6.3.3: Clock Resolution. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.13. Forensic Marking**

Step	Procedure	Yes	No	Measured Data
76.	Perform the procedure given in Section 6.4.1: FM Application Constraints. Check Yes in this row if the procedure succeeds, otherwise check No.			
77.	Perform the procedure given in Section 6.4.2: Granularity of FM Control. Check Yes in this row if the procedure succeeds, otherwise check No.			
78.	Perform the procedure given in Section 6.4.3: FM Payload. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.14. Secure Processing Block Type 2**

Step	Procedure	Yes	No	Measured Data
79.	Perform the procedure given in Section 5.1.2: SPB Type 2 Security Perimeter. Check Yes in this row if the procedure succeeds, otherwise check No.			
80.	Perform the procedure given in Section 5.1.3: SPB Type 2 Secure Silicon. Check Yes in this row if the procedure succeeds, otherwise check No.			
81.	Perform the procedure given in Section 7.2.3: SPB2 Requirements. Check Yes in this row if the procedure succeeds, otherwise check No.			
82.	Perform the procedure given in Section 7.2.4: SPB2 Secure Silicon Requirements. Check Yes in this row if the procedure succeeds, otherwise check No.			
83.	Perform the procedure given in Section 7.2.5: SPB2 Tamper Evidence. Check Yes in this row if the procedure succeeds, otherwise check No.			
84.	Perform the procedure given in Section 7.2.6: SPB2 Secure Silicon Field Replacement. Check Yes in this row if the procedure succeeds, otherwise check No.			



Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
85.	Perform the procedure given in Section 7.2.1: Projector Physical Protection. Check Yes in this row if the procedure succeeds, otherwise check No.			
86.	Perform the procedure given in Section 7.3.1: Projector Companion SPB Location. Check Yes in this row if the procedure succeeds, otherwise check No.			
87.	Perform the procedure given in Section 7.2.7: Systems Without Electronic Marriage. Check Yes in this row if the procedure succeeds, otherwise check No.			
88.	Perform the procedure given in Section 7.2.8: Electronic Marriage Break Key Retaining. Check Yes in this row if the procedure succeeds, otherwise check No.			
89.	Perform the procedure given in Section 7.2.2: Projector Access Door. Check Yes in this row if the procedure succeeds, otherwise check No.			

**Table 15.15. Image Processing**

Step	Procedure	Yes	No	Measured Data
90.	Perform the procedure given in Section 7.5.1: Projector Overlay. Check Yes in this row if the procedure succeeds, otherwise check No.			
91.	Perform the procedure given in Section 7.5.2: Projector Lens. Check Yes in this row if the procedure succeeds, otherwise check No.			
92.	Perform the procedure given in Section 7.5.3: Projector Pixel Count/Structure. Check Yes in this row if the procedure succeeds, otherwise check No.			
93.	Perform the procedure given in Section 7.5.4: Projector Spatial Resolution and Frame Rate Conversion. Check Yes in this row if the procedure succeeds, otherwise check No.			
94.	Perform the procedure given in Section 7.5.5: White Point Luminance and Uniformity. Check Yes in this row if the procedure succeeds, otherwise check No.			
95.	Perform the procedure given in Section 7.5.6: White Point Chromaticity and Uniformity. Check Yes in this row if the procedure succeeds, otherwise check No.			
96.	Perform the procedure given in Section 7.5.7: Sequential Contrast. Check Yes in this row if the procedure succeeds, otherwise check No.			
97.	Perform the procedure given in Section 7.5.8: Intra-frame Contrast. Check Yes in this row if the procedure succeeds, otherwise check No.			

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

Step	Procedure	Yes	No	Measured Data
98.	Perform the procedure given in Section 7.5.9: Grayscale Tracking. Check Yes in this row if the procedure succeeds, otherwise check No.			
99.	Perform the procedure given in Section 7.5.10: Contouring. Check Yes in this row if the procedure succeeds, otherwise check No.			
100.	Perform the procedure given in Section 7.5.11: Transfer Function. Check Yes in this row if the procedure succeeds, otherwise check No.			
101.	Perform the procedure given in Section 7.5.12: Color Accuracy. Check Yes in this row if the procedure succeeds, otherwise check No.			

# Appendix A. Test Materials

## A.1. Overview

To facilitate consistent testing of d-cinema equipment, a set of reference files has been produced to be used as directed in the respective test procedures. These materials are described in detail in this Appendix with the intention that the materials can be re-created from the descriptions and used to achieve testing results equivalent to those achieved with the original reference files.

The test material described below consists of digital certificates, Key Delivery Messages (KDM) and D-Cinema Packages (DCP). A DCP can be further deconstructed as a set of Track Files, Composition Playlists and related file descriptions. Some Track Files will be encrypted.

Because the identity of a Test Subject cannot be known until the device has been manufactured, it is not possible to create reference KDM files in advance. It is therefore necessary to divide the test material into two categories: common-use reference material and per-device reference material. Common-use reference material can be created once and used without limit on any compliant system. Per-device reference material must be created for each Test Subject, with foreknowledge of the date and time of the test session.

Two additional categories of reference material exist: compliant and intentionally non-compliant. Most of the material will be "golden" reference files, intended to be entirely compliant with the relevant specifications. Other files, however, will be intentionally broken to allow testing of error detection and recovery mechanisms.

## A.2. Images

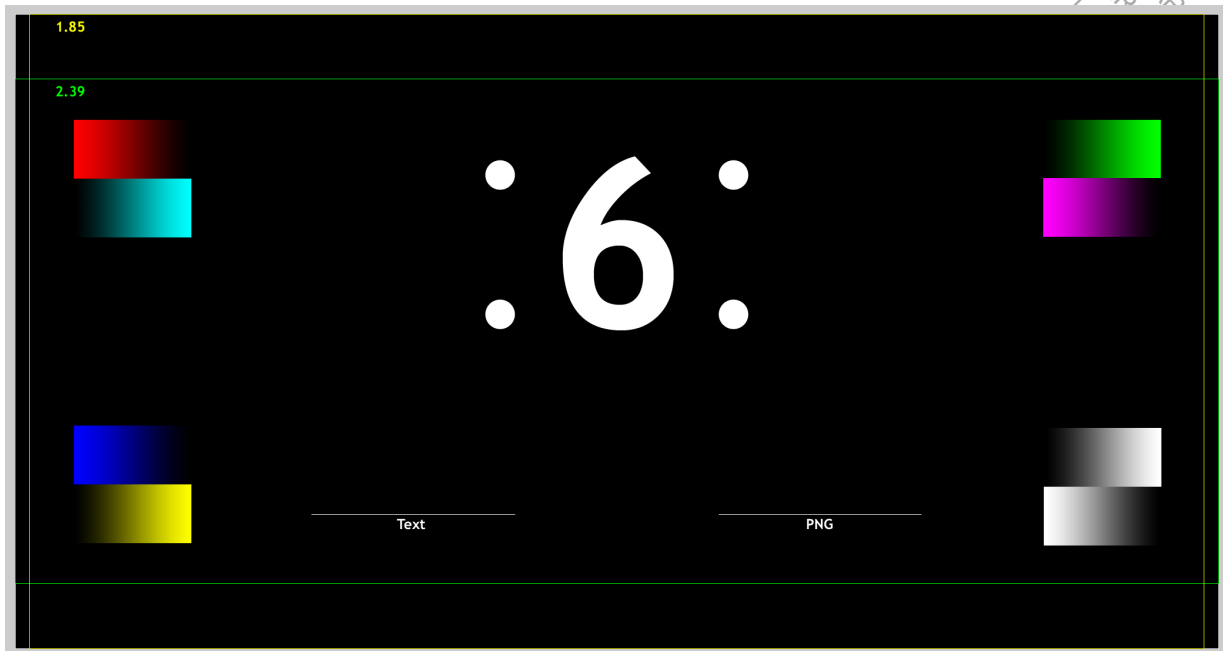
### A.2.1. Introduction

This section defines a set of MXF track files. For each track file, a description is given which details the images encoded in the file. The image track files will be combined with sound files to make complete compositions (see Section A.4).

### A.2.2. Sync Count

Item Data	Data Description
type:	MXF j2c
filename:	sync_count_j2c_pt.mxf
description:	MXF track file containing two seconds (48 frames) of full black frames followed by a ten second countdown and two seconds of black. The countdown consists of ten identical one-second count segments, from 9-0. Each count segment consists of twenty-four frames of the respective digit for the count period. The first frame of each count segment will have a punch set to indicate sync. The example image below shows the first frame of the fourth count period, which contains the number 6 (six).
conforms to:	S377M-2004, S429-3-2006, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:14
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Figure A.1. Sync Count



### A.2.3. Sync Count (Encrypted)

Item Data	Data Description
type:	MXF j2c
filename:	sync_count_j2c_ct.mxf
description:	Encrypted MXF track file, contents are identical to sync_count_j2c_pt.
conforms to:	S377M-2004, S429-3-2006, S429-6-2006, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:14
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

### A.2.4. 4K sync count

Item Data	Data Description
type:	MXF j2c
filename:	4K_sync_count_j2c_pt.mxf
description:	
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None

## A.2.5. Sync Count 48fps

Item Data	Data Description
type:	MXF j2c
filename:	sync_count_48fps_j2c_pt.mxf
description:	
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None

## A.2.6. Channel I.D. 5.1

Item Data	Data Description
type:	MXF j2c
filename:	channel_id_51_j2c_pt.mxf
description:	MXF track file containing two seconds (48 frames) of full black frames followed by a thirty second audio channel identification set and two seconds of black. The audio channel identification set consists of six identical five-second identifier segments having the following consecutively displayed labels: Left, Center, Right, Left Surround, Right Surround, LFE. Each channel identifier segment consists of five seconds (120 frames) of the respective label. The example image below shows the first frame of the first label period, which contains the label "Left".
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:34
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Version 1.0 AP  
 NEWER VERSIONS  
 Version 1.1  
 FOR INTERNAL USE ONLY  
 UNAVAILABLE  
 2009

Figure A.2. Channel I.D. 5.1



### A.2.7. Channel I.D. 7.1

Item Data	Data Description
type:	MXF j2c
filename:	channel_id_71_j2c_pt.mxf
description:	MXF track file containing two seconds (48 frames) of full black frames followed by a forty second audio channel identification set and two seconds of black. The audio channel identification set consists of eight identical five-second identifier segments having the following consecutively displayed labels: Left, Left Center, Center, Right Center, Right, Left Surround, Right Surround, LFE. Each channel identifier segment consists of five seconds (120 frames) of the respective label.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:44
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

### A.2.8. Channel I.D. 1-16

Item Data	Data Description
type:	MXF j2c
filename:	channel_id_01-16_j2c_pt.mxf
description:	MXF track file containing two seconds (48 frames) of full black frames followed by an eighty second audio channel identification set and two seconds of black. The audio channel identification set consists of sixteen identical five-second identifier segments displaying consecutively numbered channel labels: 1, 2, 3, 4, etc. through 16. Each channel identifier segment consists of five seconds (120 frames) of the respective label. The example image below shows the first frame of the first label period, which contains the label "1".
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	1:24
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

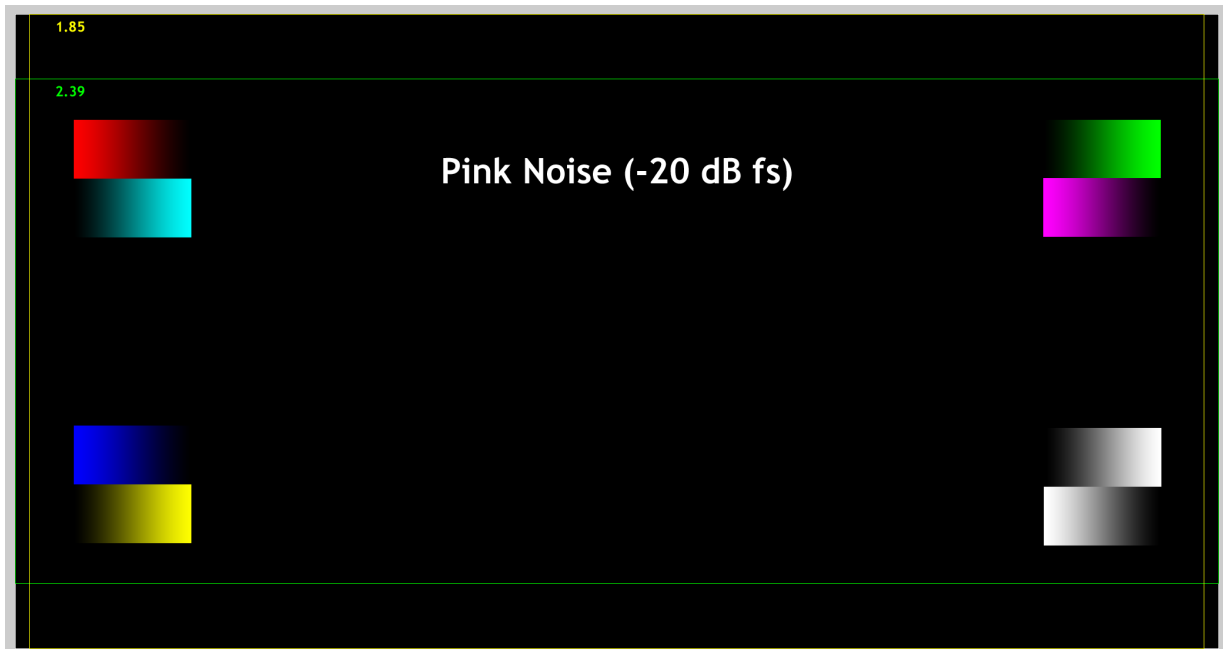
Figure A.3. Channel I.D. 1-16



## A.2.9. Pink Noise 7.1

Item Data	Data Description
type:	MXF j2c
filename:	pink_noise_71_j2c_pt.mxf
description:	MXF track file containing 30 seconds (720 frames) of the text "Pink Noise (-20 dBFS)".
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Figure A.4. Pink Noise 7.1

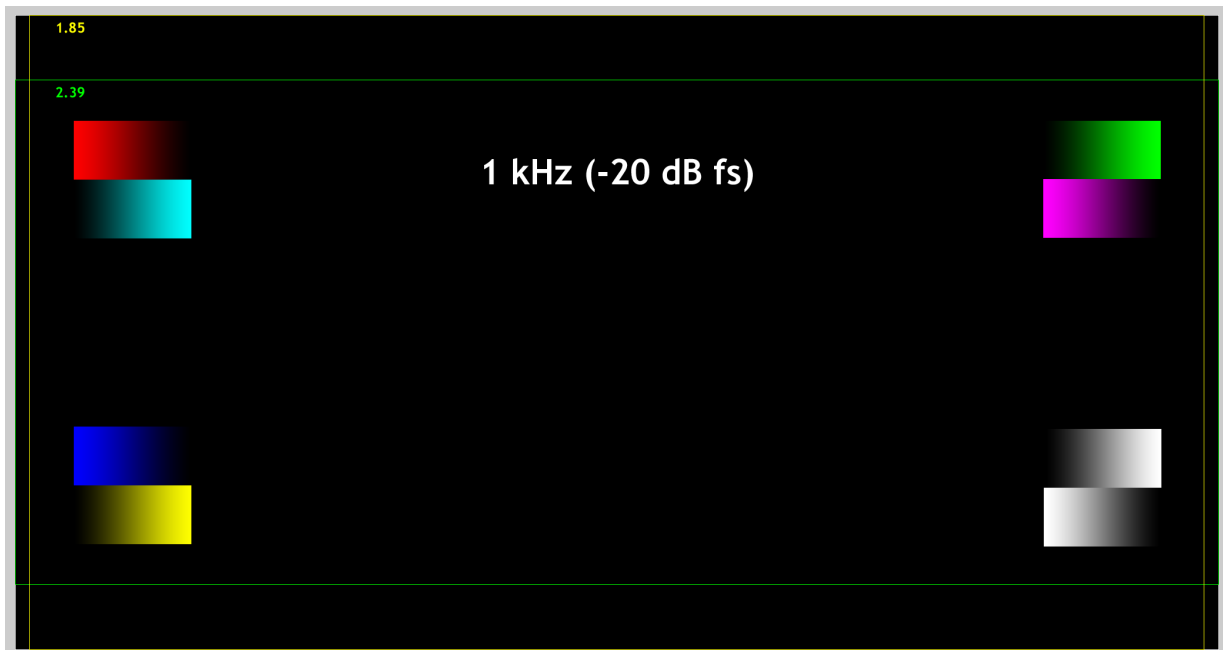




## A.2.10. 1 kHz Sine Wave

Item Data	Data Description
type:	MXF j2c
filename:	1_khz_sine_wave_j2c_pt.mxf
description:	MXF track file containing 30 seconds (720 frames) of the text "1 kHz Sine Wave (-20 dBFS)".
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Figure A.5. 1 kHz Sine Wave



Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

### A.2.11. 400 Hz Sine Wave

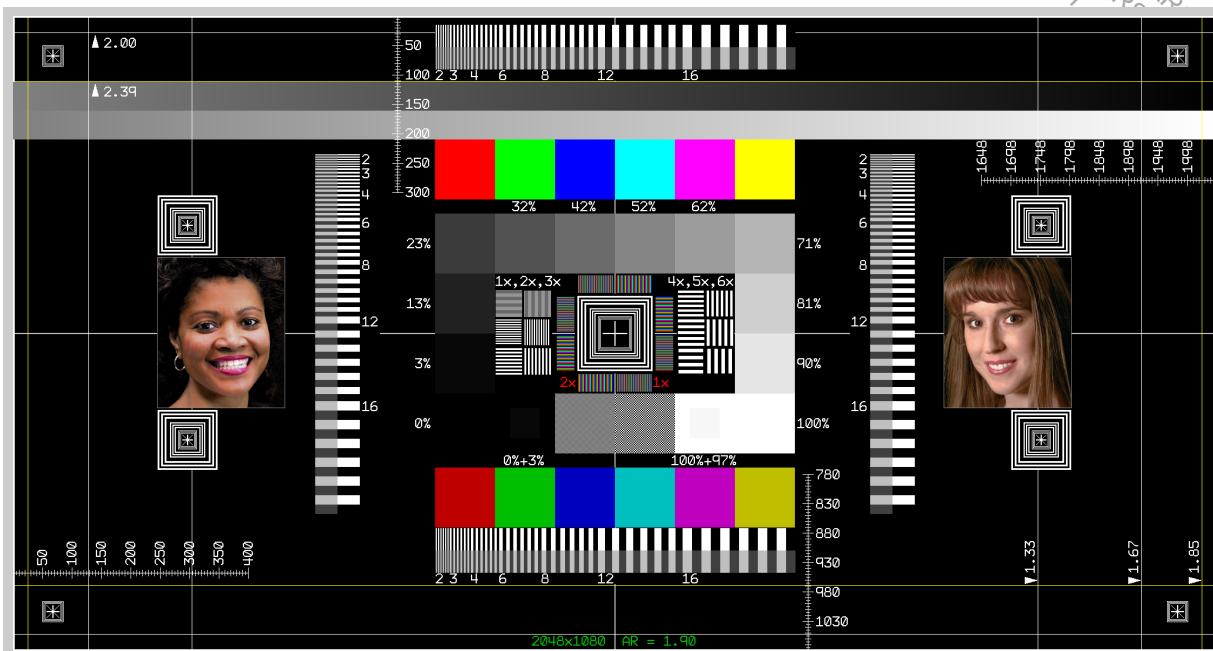
Item Data	Data Description
type:	MXF j2c
filename:	400_hz_sine_wave_j2c_pt.mxf
description:	MXF track file containing 30 seconds (720 frames) of the text "400 Hz Sine Wave (-20 dBFS)".
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

### A.2.12. "NIST" 2k Test Pattern

Item Data	Data Description
type:	MXF j2c
filename:	nist_2k_test_pattern_j2c_pt.mxf
description:	MXF track file containing the "DCI NIST" pattern created during original DCI research project. The pattern (shown below) includes geometric dimensions, color chips, dimensional patterns, a grayscale gradient and full-color photographic images. The track file contains 30 seconds (720 frames) of this image.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Version 1.0 AR  
NEWER VERSION AVAILABLE  
Version 1.1

Figure A.6. "NIST" 2k Test Pattern



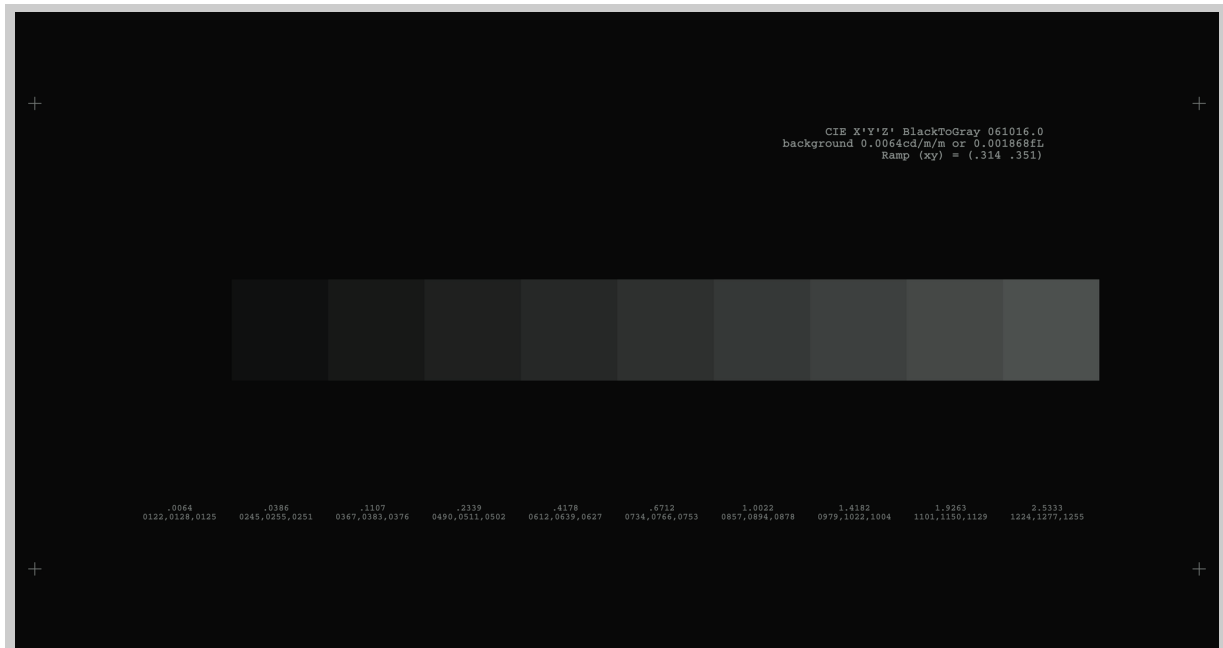
### A.2.13. "NIST" 4k Test Pattern

Item Data	Data Description
type:	MXF j2c
filename:	nist_4k_test_pattern_j2c_pt.mxf
description:	MXF track file containing the "DCI NIST" shown in Section A.2.12. The track file contains 30 seconds (720 frames) of this image.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: PixelArraySize	4096x2160
meta: EditRate	24/1

## A.2.14. Black to Gray Step Series

Item Data	Data Description
type:	MXF j2c
filename:	gray_step_j2c_pt.mxf
description:	MXF track file containing five seconds (120 frames) of a chart showing all gray step values for the Black to Gray values in Section 7.5.9: Grayscale Tracking. This is followed by 5 seconds of each of the 10 values as a full frame.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:55
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Figure A.7. Black to Gray Step Series



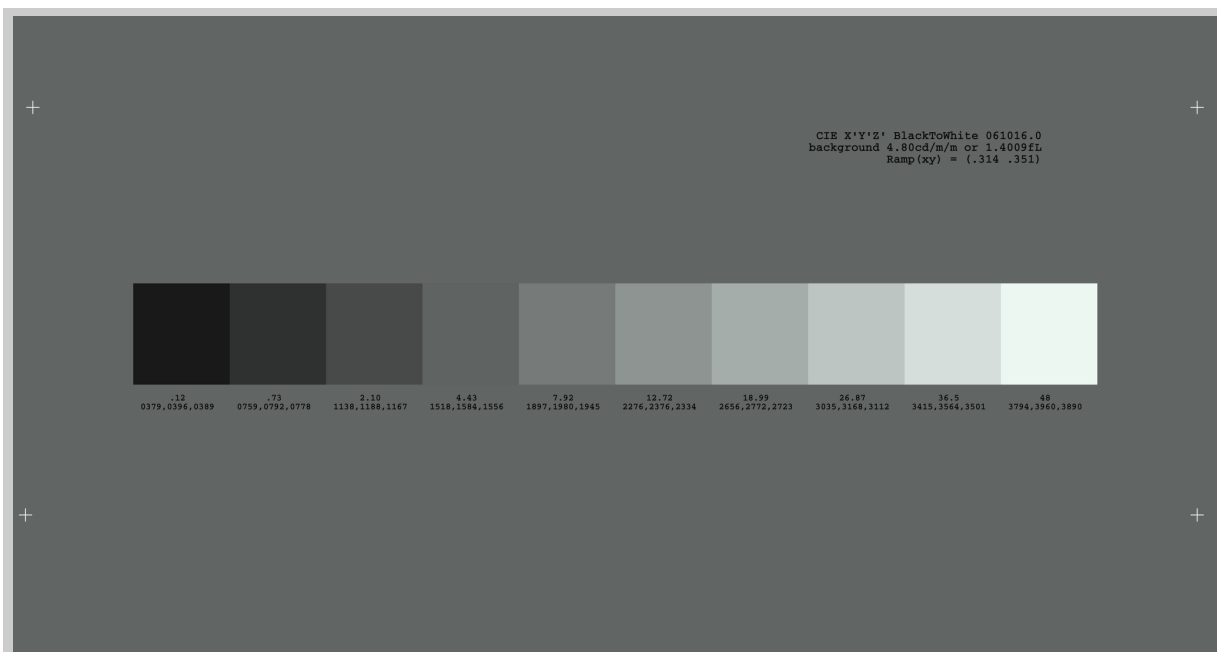
## A.2.15. 4K gray step

Item Data	Data Description
type:	MXF j2c
filename:	4K_gray_step_j2c_pt.mxf
description:	
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None

## A.2.16. Black to White Step Series

Item Data	Data Description
type:	MXF j2c
filename:	white_step_j2c_pt.mxf
description:	MXF track file containing five seconds (120 frames) of a chart showing all gray step values for the Black to White values in Section 7.5.9: Grayscale Tracking. This is followed by 5 seconds of each of the 10 values as a full frame.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:55
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Figure A.8. Black to White Step Series



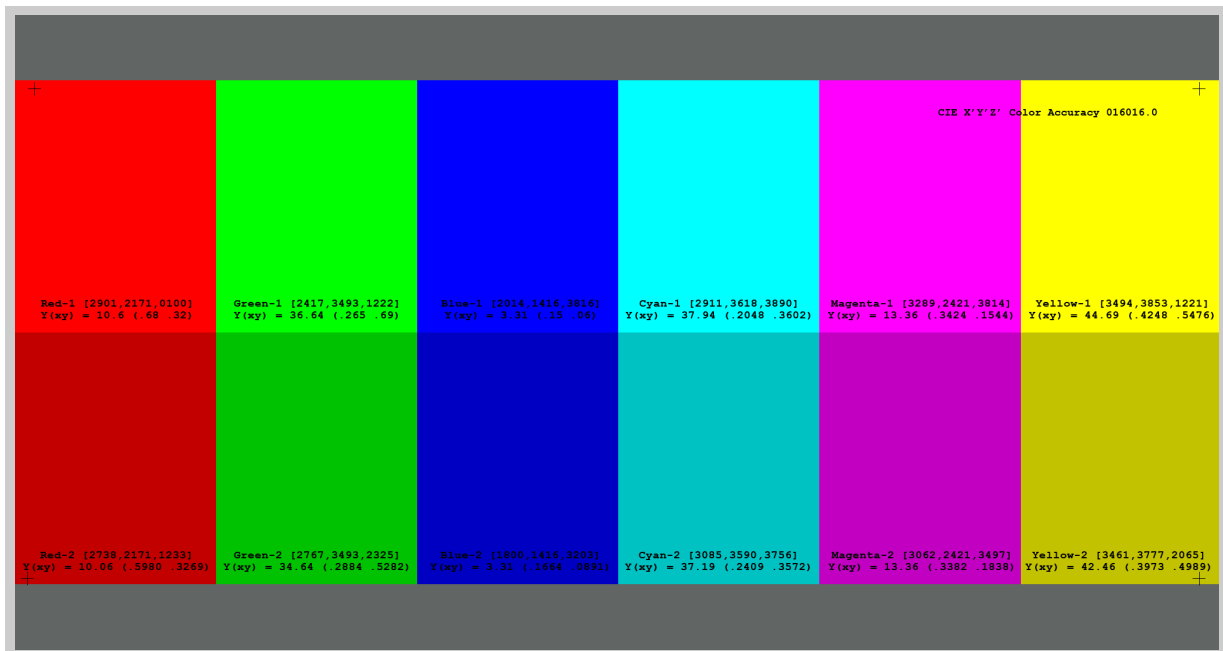
## A.2.17. 4K grayscale grad

Item Data	Data Description
type:	MXF j2c
filename:	4K_grayscale_grad_j2c_pt.mxf
description:	
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None

## A.2.18. Color Accuracy Series

Item Data	Data Description
type:	MXF j2c
filename:	color_accuracy_j2c_pt.mxf
description:	MXF track file containing five seconds (120 frames) of a chart showing all color values for the test in Section 7.5.12: Color Accuracy. This is followed by 5 seconds of each of the 12 color values as a full frame.
conforms to:	S377M-2004, S429-3-2006, S422M
prerequisites:	None
malformations:	None
meta: Duration	1:05
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

Figure A.9. Color Accuracy Series



## A.2.19. Contouring

Item Data	Data Description
type:	MXF j2c
filename:	contouring_j2c_pt.mxf
description:	MXF track file containing images with gradients to reveal contouring artifacts.
conforms to:	
prerequisites:	None
malformations:	None
meta: Duration	
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

## A.2.20. Black (Empty Frame)

Item Data	Data Description
type:	MXF j2c
filename:	black_j2c_pt.mxf
description:	MXF track file containing 30 seconds (720 frames) of black (all pixels zero).
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

## A.2.21. White (White Frame)

Item Data	Data Description
type:	MXF j2c
filename:	white_j2c_pt.mxf
description:	MXF track file containing 30 seconds (720 frames) of white.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

## A.2.22. Checkerboard Frame

Item Data	Data Description
type:	MXF j2c
filename:	2K_checkerboard_j2c_pt.mxf
description:	MXF track file containing 30 seconds (720 frames) of checkerboard pattern.
conforms to:	
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

## A.2.23. Frame Number Burn-in

Item Data	Data Description
type:	MXF j2c
filename:	frame_number_burn_in_j2c_pt.mxf
description:	MXF track file containing a sequence of frames having a visible number field. The six digit field contains 000000 in the first frame of the file, with each consecutive frame increasing the count by 1. The last frame will be numbered 007199.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	5:00
meta: PixelArraySize	2048x1080
meta: EditRate	24/1

## A.2.24. Frame Number Burn-in, 48fps

Item Data	Data Description
type:	MXF j2c
filename:	frame_number_burn_in_48fps_j2c_pt.mxf
description:	MXF track file containing a sequence of frames having a visible number field. The six digit field contains 000000 in the first frame of the file, with each consecutive frame increasing the count by 1. The last frame will be numbered 014399.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None
meta: Duration	5:00
meta: PixelArraySize	2048x1080
meta: EditRate	48/1



**A.2.25. StEM 2K**

Item Data	Data Description
type:	MXF j2c
filename:	StEM_2K_j2c_pt.mxf
description:	MXF track file containing the complete DCI StEM Mini Movie.
conforms to:	S377M-2004, S429-3-2006, S429-6-2006, S422M
prerequisites:	None
malformations:	None
meta: Duration	12:05
meta: PixelArraySize	2048x858
meta: EditRate	24/1

**A.2.26. StEM 2K (Encrypted)**

Item Data	Data Description
type:	MXF j2c
filename:	StEM_2K_j2c_ct.mxf
description:	Encrypted MXF track file containing the complete DCI StEM Mini Movie.
conforms to:	S377M-2004, S429-3-2006, S429-6-2006, S422M
prerequisites:	None
malformations:	None
meta: Duration	12:05
meta: PixelArraySize	2048x858
meta: EditRate	24/1

**A.2.27. StEM 4K**

Item Data	Data Description
type:	MXF j2c
filename:	StEM_4K_j2c_pt.mxf
description:	MXF track file containing the complete DCI StEM Mini Movie.
conforms to:	S377M-2004, S429-3-2006, S429-6-2006, S422M
prerequisites:	None
malformations:	None
meta: Duration	12:05
meta: PixelArraySize	4096x1716
meta: EditRate	24/1

**A.2.28. StEM 4K (Encrypted)**

Item Data	Data Description
type:	MXF j2c
filename:	StEM_4K_j2c_ct.mxf
description:	Encrypted MXF track file containing the complete DCI StEM Mini Movie.
conforms to:	S377M-2004, S429-3-2006, S429-6-2006, S422M
prerequisites:	None
malformations:	None
meta: Duration	12:05
meta: PixelArraySize	4096x1716
meta: EditRate	24/1

**A.2.29. m01 pict foos**

Item Data	Data Description
type:	MXF j2c
filename:	m01_pict_foos_j2c_pt.mxf
description:	
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None

**A.2.30. m03 snd splc**

Item Data	Data Description
type:	MXF j2c
filename:	m03_snd_splc_j2c_pt.mxf
description:	
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	None

**A.2.31. m09 pict bad hmac**

Item Data	Data Description
type:	MXF j2c
filename:	m09_pict_bad_hmac_j2c_pt.mxf
description:	Picture track file in which one of the HMAC values for a single frame has been changed.
conforms to:	S377M-2004, S429-3-2006, S429-4, S422M
prerequisites:	None
malformations:	TDB

## A.3. Sound

### A.3.1. Introduction

This section defines a set of sound elements and MXF track files. The MXF track files will be used in Section A.4.

### A.3.2. Sync Count 5.1

Item Data	Data Description
type:	MXF pcm
filename:	sync_count_51_pcm_pt.mxf
description:	MXF track file containing six channels of audio. Channels 1,2,4,5 and 6 are silent. Channel 3 (Center) contains two seconds (48 frames) of silence followed by a ten second countdown and two seconds of silence. The countdown consists of ten identical one-second count segments. Each count segment consists of one frame (2000 samples) encoding a 1 kHz sine wave at -20 dBFS, followed by 23 frames of silence.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	0:14
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

### A.3.3. Sync Count 5.1 (Encrypted)

Item Data	Data Description
type:	MXF pcm
filename:	sync_count_51_pcm_ct.mxf
description:	Encrypted MXF track file, contents are identical to sync_count_51_pcm_pt.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	0:14
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

### A.3.4. Sync Count 5.1 48fps

Item Data	Data Description
type:	MXF pcm
filename:	sync_count_51_48fps_pcm_pt.mxf
description:	MXF track file, contents are identical to sync_count_51_pcm_pt. Note that 48 fps frame rate. Because of this, each count segment consists of two frames (1000 samples) encoding a 1 kHz sine wave at -20 dBFS, followed by 46 frames of silence.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	0:14
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	48/1

### A.3.5. Channel I.D. 5.1

Item Data	Data Description
type:	MXF pcm
filename:	channel_id_51_pcm_pt.mxf
description:	MXF track file containing a repeated voice announcement of the channel label of the respective channel: Left, Center, Right, Left Surround, Right Surround, LFE. Voice announcements are sequential, in the order given here.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

### A.3.6. Channel I.D. 7.1

Item Data	Data Description
type:	MXF pcm
filename:	channel_id_71_pcm_pt.mxf
description:	MXF track file containing a repeated voice announcement of the channel label of the respective channel: Left, Left Center, Center, Right, Right Center, Left Surround, Right Surround, LFE. Voice announcements are sequential, in the order given here.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: SampleRate	48000
meta: SoundFormat	7.1
meta: EditRate	24/1

### A.3.7. Channel I.D. 1-16

Item Data	Data Description
type:	MXF pcm
filename:	channel_id_01-16_pcm_pt.mxf
description:	MXF track file containing a repeated voice announcement of the channel number of the respective channel. Announcements occur simultaneously on all channels.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: SampleRate	48000
meta: SoundFormat	16
meta: EditRate	24/1

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8, 2009

### A.3.8. Pink Noise, 16 Channels

Item Data	Data Description
type:	MXF pcm
filename:	pink_noise_71_pcm_pt.mxf
description:	Pink ( $1/f$ ) noise at -20 dBFS on sixteen channels, band limited to 22 KHz. Identical, sample-aligned signal must be used on all channels.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	0:30
meta: SampleRate	48000
meta: SoundFormat	16
meta: EditRate	24/1

### A.3.9. Pink Noise, 16 Channels, 96 kHz

Item Data	Data Description
type:	MXF pcm
filename:	pink_noise_71_96khz_pcm_pt.mxf
description:	Pink ( $1/f$ ) noise at -20 dBFS on sixteen channels, band limited to 44 KHz. Identical, sample-aligned signal must be used on all channels.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	1:00
meta: SampleRate	96000
meta: SoundFormat	16
meta: EditRate	24/1

**A.3.10. 1 kHz Sine Wave**

Item Data	Data Description
type:	MXF pcm
filename:	1_khz_sine_wave_pcm_pt.mxf
description:	1 kHz sine wave on sixteen channels at -20 dBFS.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	1:00
meta: SampleRate	48000
meta: SoundFormat	16
meta: EditRate	24/1

**A.3.11. 1 kHz Sine Wave**

Item Data	Data Description
type:	MXF pcm
filename:	1_khz_sine_wave_96khz_pcm_pt.mxf
description:	1 kHz sine wave on sixteen channels at -20 dBFS.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	1:00
meta: SampleRate	96000
meta: SoundFormat	16
meta: EditRate	24/1

**A.3.12. 400 hz sine wave**

Item Data	Data Description
type:	MXF pcm
filename:	400_hz_sine_wave_pcm_pt.mxf
description:	400 Hz sine wave on six channels at -20 dBfs (dB Full Scale). LFE channel is full-range.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	1:00
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

**A.3.13. Silence, 5.1**

Item Data	Data Description
type:	MXF pcm
filename:	black_51_pcm_pt.mxf
description:	MXF track file containing six channels of silence.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	2:00
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

**A.3.14. Silence, 5.1 (Encrypted)**

Item Data	Data Description
type:	MXF pcm
filename:	black_51_pcm_ct.mxf
description:	Encrypted MXF track file containing six channels of silence.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	2:00
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

**A.3.15. StEM 5.1 Sound**

Item Data	Data Description
type:	MXF pcm
filename:	StEM_51_pcm_pt.mxf
description:	MXF track file containing 5.1 sound essence for the DCI StEM Mini Movie.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	12:05
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1



**A.3.16. StEM 5.1 Sound (Encrypted)**

Item Data	Data Description
type:	MXF pcm
filename:	StEM_51_pcm_ct.mxf
description:	Encrypted MXF track file containing 5.1 sound essence for the DCI StEM Mini Movie.
conforms to:	S377M-2004, S429-3-2006, S429-6-2006, S382M-2006
prerequisites:	None
malformations:	None
meta: Duration	12:05
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

**A.3.17. m02 snd foos**

Item Data	Data Description
type:	MXF pcm
filename:	m02_snd_foos_pcm_pt.mxf
description:	Broken MXF track file containing six channels of silence.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	The KLV packets containing edit units 96 and 97 are swapped.
meta: Duration	0:30
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

**A.3.18. m10 snd bad hmac**

Item Data	Data Description
type:	MXF pcm
filename:	m10_snd_foos_pcm_pt.mxf
description:	Sound track file in which one of the HMAC values for a single frame has been changed.
conforms to:	S377M-2004, S429-3-2006, S382M-2006
prerequisites:	None
malformations:	TBD
meta: Duration	0:30
meta: SampleRate	48000
meta: SoundFormat	5.1
meta: EditRate	24/1

## A.4. D-Cinema Packages

### A.4.1. Introduction

This section defines a set of D-Cinema Compositions and D-Cinema Packages. The Compositions depend upon the track files created in Section A.2 and Section A.3. The Packages contain the Compositions for ingest.

### A.4.2. DCI 2K Sync Test

Item Data	Data Description
type:	CPL
filename:	sync_test.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	sync_count_j2c_pt.mxf
malformations:	None

### A.4.3. DCI 2K Sync test with Subtitles

Item Data	Data Description
type:	CPL
filename:	sync_test_with_subs.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	sync_count_j2c_pt.mxf
malformations:	None

### A.4.4. DCI 2K Sync test with Subtitles (Encrypted)

Item Data	Data Description
type:	CPL
filename:	sync_test_with_subs_ct.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	sync_count_j2c_ct.mxf
malformations:	None

### A.4.5. DCI 2K Sync Test (48fps)

Item Data	Data Description
type:	CPL
filename:	sync_test_48fps.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	sync_count_48fps_j2c_pt.mxf
malformations:	None

### A.4.6. 4K Sync Test

Item Data	Data Description
type:	CPL
filename:	4K_sync_test.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	4K_sync_count_j2c_pt.mxf
malformations:	None

### A.4.7. DCI 5.1 Channel Identification

Item Data	Data Description
type:	CPL
filename:	channel_id_51.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	channel_id_51_j2c_pt.mxf
malformations:	None

### A.4.8. 5.1 Channel Identification

Item Data	Data Description
type:	CPL
filename:	4K_channel_id_51.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	4K_channel_id_51_j2c_pt.mxf
malformations:	None

### A.4.9. DCI 7.1 Channel Identification

Item Data	Data Description
type:	CPL
filename:	channel_id_71.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	channel_id_71_j2c_pt.mxf
malformations:	None

### A.4.10. 7.1 Channel Identification

Item Data	Data Description
type:	CPL
filename:	4K_channel_id_71.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	4K_channel_id_71_j2c_pt.mxf
malformations:	None

### A.4.11. DCI 0-16 Numbered Channel Identification

Item Data	Data Description
type:	CPL
filename:	channel_id_01-16.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	channel_id_01-16_j2c_pt.mxf
malformations:	None

### A.4.12. 0-16 Numbered Channel

Item Data	Data Description
type:	CPL
filename:	4K_channel_id_01-16.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	4K_channel_id_1-16_j2c_pt.mxf
malformations:	None

### A.4.13. DCI Gray Steps

Item Data	Data Description
type:	CPL
filename:	gray_step.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	gray_step_j2c_pt.mxf
malformations:	None

### A.4.14. DCI White Steps

Item Data	Data Description
type:	CPL
filename:	white_step.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	white_step_j2c_pt.mxf
malformations:	None

### A.4.15. DCI Grayscale Gradient

Item Data	Data Description
type:	CPL
filename:	gray_scale_gradient.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	gray_scale_grad_j2c_pt.mxf
malformations:	None

### A.4.16. 4K Gray Steps

Item Data	Data Description
type:	CPL
filename:	4K_gray_step.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	4K_gray_step_j2c_pt.mxf
malformations:	None

### A.4.17. 4K Grayscale Gradient

Item Data	Data Description
type:	CPL
filename:	4K_gray_scale_gradient.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	4K_grayscale_grad_j2c_pt.mxf
malformations:	None

### A.4.18. Color Accuracy Series

Item Data	Data Description
type:	CPL
filename:	color_accuracy_pt.cpl.xml
description:	Composition containing five seconds (120 frames) of a chart showing all color values for the test in Section 7.5.12: Color Accuracy. This is followed by 5 seconds of each of the 12 color values as a full frame.
conforms to:	S377M-2004, S429-3-2006, S422M
prerequisites:	None
malformations:	None

### A.4.19. Contouring Sequence

Item Data	Data Description
type:	CPL
filename:	Contouring.cpl.xml
description:	Composition containing a sequence of images with gradients to reveal contouring artifacts.
conforms to:	
prerequisites:	contouring_j2c_pt.mxf
malformations:	None

### A.4.20. DCI NIST Frame with silence

Item Data	Data Description
type:	CPL
filename:	nist_pattern_black_audio.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	nist_2k_test_pattern_j2c_pt.mxf
malformations:	None

**A.4.21. DCI NIST Frame with Pink Noise**

Item Data	Data Description
type:	CPL
filename:	nist_pattern_pink_noise.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	nist_2k_test_pattern_j2c_pt.mxf
malformations:	None

**A.4.22. DCI NIST Frame with 1 kHz tone (-20 dB fs)**

Item Data	Data Description
type:	CPL
filename:	nist_pattern_1k.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	nist_2k_test_pattern_j2c_pt.mxf
malformations:	None

**A.4.23. DCI NIST Frame with Pink Noise (96 kHz)**

Item Data	Data Description
type:	CPL
filename:	nist_pattern_pink_noise_96k.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	nist_2k_test_pattern_j2c_pt.mxf
malformations:	None

**A.4.24. DCI NIST Frame with 1 kHz tone (-20 dB fs, 96kHz)**

Item Data	Data Description
type:	CPL
filename:	nist_pattern_1k_96k.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	nist_2k_test_pattern_j2c_pt.mxf
malformations:	None

**A.4.25. DCI NIST Frame no sound files**

Item Data	Data Description
type:	CPL
filename:	nist_pattern_no_audio.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	nist_2k_test_pattern_j2c_pt.mxf
malformations:	None

**A.4.26. 4K NIST Pattern**

Item Data	Data Description
type:	CPL
filename:	4K_nist_pattern.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	4K_nist_test_pattern_j2c.mxf
malformations:	None

**A.4.27. DCI 2K Image with Frame Number Burn In**

Item Data	Data Description
type:	CPL
filename:	frame_num_burn_in.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	frame_number_burn_in_j2c_pt.mxf
malformations:	None

**A.4.28. DCI 2K Image with Frame Number Burn In (48 fps)**

Item Data	Data Description
type:	CPL
filename:	frame_num_burn_in_48fps.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	frame_number_burn_in_48fps_j2c_pt.mxf
malformations:	None



### A.4.29. DCI 2K Image with Frame Number Burn In (Flat)

Item Data	Data Description
type:	CPL
filename:	frame_count_flat_2_reels.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	TBD
malformations:	None

### A.4.30. DCI 2K Image with Frame Number Burn In (Scope)

Item Data	Data Description
type:	CPL
filename:	frame_count_scope_2_reels.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	TBD
malformations:	None

### A.4.31. DCI 2K StEM Test Sequence

Item Data	Data Description
type:	CPL
filename:	2K_StEM_sequence_pt.cpl.xml
description:	A plaintext composition consisting of six (6) reels. Each reel is composed of the StEM 2K image and 5.1 sound track files.
conforms to:	S429-7-2006
prerequisites:	StEM_2K_j2c_pt.mxf StEM_51_pcm_pt.mxf
malformations:	None

### A.4.32. DCI 2K StEM Test Sequence (Encrypted)

Item Data	Data Description
type:	CPL
filename:	2K_StEM_sequence_ct.cpl.xml
description:	An encrypted composition consisting of six (6) reels. Each reel is composed of the encrypted StEM 2K image and encrypted 5.1 sound track files.
conforms to:	S429-7-2006
prerequisites:	StEM_2K_j2c_ct.mxf, StEM_51_pcm_ct.mxf
malformations:	None

**A.4.33. 4K StEM Test Sequence**

Item Data	Data Description
type:	CPL
filename:	4K_StEM_sequence_pt.cpl.xml
description:	A plaintext composition consisting of six (6) reels. Each reel is composed of the StEM 4K image and 5.1 sound track files.
conforms to:	S429-7-2006
prerequisites:	StEM_4K_j2c_pt.mxf StEM_51_pcm_pt.mxf
malformations:	None

**A.4.34. 4K StEM Test Sequence (Encrypted)**

Item Data	Data Description
type:	CPL
filename:	4K_StEM_sequence_ct.cpl.xml
description:	An encrypted composition consisting of six (6) reels. Each reel is composed of the encrypted StEM 4K image and encrypted 5.1 sound track files.
conforms to:	S429-7-2006
prerequisites:	StEM_4K_j2c_ct.mxf StEM_51_pcm_ct.mxf
malformations:	None

**A.4.35. 128 Reel Composition, "A" Series (Encrypted)**

Item Data	Data Description
type:	CPL
filename:	2K_StEM_128_a_reels_ct.cpl.xml
description:	An encrypted composition consisting of one hundred and twenty eight (128) reels. Each reel is composed of part of the encrypted StEM 2K image and encrypted 5.1 sound track files.
conforms to:	S429-7-2006
prerequisites:	StEM_2K_j2c_ct.mxf, StEM_51_pcm_ct.mxf
malformations:	None

### A.4.36. 128 Reel Composition, "B" Series (Encrypted)

Item Data	Data Description
type:	CPL
filename:	2K_StEM_128_b_reels_ct.cpl.xml
description:	An encrypted composition consisting of one hundred and twenty eight (128) reels. Each reel is composed of part of the encrypted StEM 2K image and encrypted 5.1 sound track files.
conforms to:	S429-7-2006
prerequisites:	StEM_2K_j2c_ct.mxf, StEM_51_pcm_ct.mxf
malformations:	None

### A.4.37. DCI Black Spacer - 5 seconds

Item Data	Data Description
type:	CPL
filename:	black_spacer_5s.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	black_j2c_pt.mxf
malformations:	None

### A.4.38. White Frame Sequence

Item Data	Data Description
type:	CPL
filename:	white_pt.cpl.xml
description:	
conforms to:	S431-1-2006
prerequisites:	white_j2c_pt.mxf
malformations:	None

### A.4.39. Checkerboard Sequence

Item Data	Data Description
type:	CPL
filename:	checkerboard_pt.cpl.xml
description:	
conforms to:	S431-2
prerequisites:	2K_checkerboard_j2c_pt.mxf
malformations:	None

#### A.4.40. DCI Malformed Test 1: Picture with Frame-out-of-order error

Item Data	Data Description
type:	CPL
filename:	m01_pict_frame_oo.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	m01_pict_foos_j2c_pt.mxf
malformations:	None

#### A.4.41. DCI Malformed Test 2: Sound with Frame-out-of-order error

Item Data	Data Description
type:	CPL
filename:	m02_snd_frame_oo.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	frame_number_burn_in_j2c_pt.mxf
malformations:	None

#### A.4.42. DCI Malformed Test 3: Sound Splice Tests

Item Data	Data Description
type:	CPL
filename:	m03_snd_splc_test.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	m03_snd_splc_j2c_pt.mxf
malformations:	None

#### A.4.43. DCI DCP With an incorrect sound file (Wrong ID)

Item Data	Data Description
type:	CPL
filename:	m04_sndtk_wrong_file.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	frame_number_burn_in_j2c_pt.mxf
malformations:	None

#### A.4.44. DCI DCP With an incorrect picture file (Wrong ID)

Item Data	Data Description
type:	CPL
filename:	m05_pict_wrong_file.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	frame_number_burn_in_j2c_pt.mxf
malformations:	None

#### A.4.45. DCI CPL with incorrect trackfile hashes

Item Data	Data Description
type:	CPL
filename:	m06_cpl_hash_error.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	sync_count_j2c_ct.mxf
malformations:	None

#### A.4.46. DCI CPL with an Invalid Signature

Item Data	Data Description
type:	CPL
filename:	m07_cpl_invalid_signature.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	frame_number_burn_in_j2c_pt.mxf
malformations:	None

#### A.4.47. DCI DCP with timed text and a missing font

Item Data	Data Description
type:	CPL
filename:	m08_dcp_timetext_missing_font.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	subtitle_background_j2c_pt.mxf
malformations:	None

Version 1.1 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8, 2009

### A.4.48. DCI Malformed Test 9: Picture with HMAC error in MXF Track File

Item Data	Data Description
type:	CPL
filename:	m09_pict_bad_hmac.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	m09_pict_bad_hmac_j2c_pt.mxf
malformations:	None

### A.4.49. DCI Malformed Test 10: Sound with HMAC error in MXF Track File

Item Data	Data Description
type:	CPL
filename:	m10_snd_bad_hmac.cpl.xml
description:	
conforms to:	S429-7-2006
prerequisites:	m10_snd_bad_hmac_pcm_pt.mxf
malformations:	None

### A.4.50. DCI DCP 2K

Item Data	Data Description
type:	DCP
filename:	DCI_2K_tests
description:	DCP containing well-formed test compositions
conforms to:	S429-8-2007,S429-9-2007
prerequisites:	sync_test, sync_test_with_subs, sync_test_with_subs_ct, sync_test_48fps, channel_id_51, channel_id_71, channel_id_01-16, KDM_success_ct, KDM_failure_ct, gray_step, gray_scale_gradient, nist_pattern_black_audio, nist_pattern_pink_noise, nist_pattern_1k, nist_pattern_pink_noise_96k, nist_pattern_1k_96k, nist_pattern_no_audio, frame_num_burn_in, frame_num_burn_in_48fps, 2K_StEM_sequence_pt, 2K_StEM_sequence_ct, subt_multiple_captions, black_spacer_5s
malformations:	None

**A.4.51. DCI DCP 2K, Malformed**

Item Data	Data Description
type:	DCP
filename:	DCI_2K_malf
description:	DCP containing malformed test compositions
conforms to:	S429-8-2007,S429-9-2007
prerequisites:	m01_pict_frame_oo, m02_snd_frame_oo, m03_snd_splc_test, m04_sndtk_wrong_file, m05_pict_wrong_file, m06_cpl_hash_error, m07_cpl_invalid_signature, m08_dcp_timetext_missing_font
malformations:	None

**A.4.52. DCI DCP 4K**

Item Data	Data Description
type:	DCP
filename:	DCI_4K_tests
description:	DCP containing well-formed 4K test compositions
conforms to:	S429-8-2007,S429-9-2007
prerequisites:	4K_sync_test, 4K_sync_test_subs, 4K_channel_id_51, 4K_channel_id_71, 4K_channel_id_01-16, 4K_nist_pattern, 4K_StEM_ct, 4K_gray_step, 4K_gray_scale_gradient, 4K_pixar_test_sequence
malformations:	None

**A.4.53. Multi-line Subtitle Test**

Item Data	Data Description
type:	CPL
filename:	multiline-caption.cpl.xml
description:	Black background with multiple-line subtitles.
conforms to:	S428-7-2006
prerequisites:	None
malformations:	None

**A.4.54. Multi-line Subtitle Test w/o Integral Font**

<b>Item Data</b>	<b>Data Description</b>
type:	CPL
filename:	multiline-caption-nofont.cpl.xml
description:	Black background with multiple-line subtitles. This package does not include the required font.
conforms to:	S428-7-2006
prerequisites:	None
malformations:	Font file missing from Timed Text Track File

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8, 2009



## A.5. Digital Certificates

Six certificate chains are defined, which separate certificates by device type and level of conformity. In the descriptions below, the IMB label refers to a certificate which contains roles for an Image Media Block or a certificate which signs such certificates. Similarly, PRJ refers to certificates or signers associated with a projector and KDS refers to certificates associated with a Key Distribution System (a KDM authoring entity).

- Chain A1 contains valid IMB certificates.
- Chain A2 contains valid IMB certificates but the chain has no intermediate signers.
- Chain A3 contains invalid IMB certificates.
- Chain B1 contains valid PRJ certificates.
- Chain C1 contains valid KDS certificates.
- Chain C3 contains invalid KDS certificates.

### A.5.1. Chain A1 IMB Certificate Files

#### A.5.1.1. chain-a1-root

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a1-root.pem
description:	Self-signed root certificate for IMB devices
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

#### A.5.1.2. chain-a1-signer1

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a1-signer1.pem
description:	Intermediate Signer, level one
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.1.3. chain-a1-signer2**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a1-signer2.pem
description:	Intermediate Signer, level two
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.2. Chain A2 IMB Certificate Files****A.5.2.1. chain-a2-root**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a2-root.pem
description:	Root cert, shallow chain
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.2.2. chain-a2-normal**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a2-normal.pem
description:	One certificate (root certificate) in chain
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3. Chain A3 IMB Certificate Files****A.5.3.1. chain-a3-root**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-root.pem
description:	Root cert, malformed leaves
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.2. chain-a3-signer1**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-signer1.pem
description:	Intermediate Signer, level one
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.3. chain-a3-osig-type**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-osig-type.pem
description:	Signature algorithm of outside signature not sha256WithRSAEncryption
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Signature algorithm of outside signature is sha1WithRSAEncryption

**A.5.3.4. chain-a3-isig-type**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-isig-type.pem
description:	Signature algorithm inside signature not sha256WithRSAEncryption
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Signature algorithm inside the signature is sha1WithRSAEncryption

**A.5.3.5. chain-a3-iosig-type**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-iosig-type.pem
description:	Signature algorithm inside and outside identical, but not sha256WithRSAEncryption
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Signature algorithm is sha1WithRSAEncryption

**A.5.3.6. chain-a3-no-rsa**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-rsa.pem
description:	Public Key not an RSA Key
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Public Key is a DSA key.

**A.5.3.7. chain-a3-short-rsa**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-short-rsa.pem
description:	Public Key Length 1024 bit
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Public key is 1024 bits.

**A.5.3.8. chain-a3-bad-exp**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-bad-exp.pem
description:	Public Key Exponent other than the required 65537
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Public Key Exponent is 43.

**A.5.3.9. chain-a3-bad-dnq**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-bad-dnq.pem
description:	dnQualifier in SubjectName field does not match RSA thumbprint
conforms to:	S430-2-2006
prerequisites:	None
malformations:	DnQualifier in Subject field is different than the calculated DnQualifier of the public key.

**A.5.3.10. chain-a3-bad-sig**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-bad-sig.pem
description:	Invalid signature
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Plaintext message digest of signature is different than the message digest of the certificate.

**A.5.3.11. chain-a3-date-ext**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-date-ext.pem
description:	Validity date of child certificate extends beyond parent certificate
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The Not After value of the leaf certificate extends 72 hours past the Not After value of the signer certificate.

**A.5.3.12. chain-a3-propext-crit**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-propext-crit.pem
description:	Contains a proprietary, critical extension
conforms to:	S430-2-2006
prerequisites:	None
malformations:	AltSubjectName is present and marked critical.

**A.5.3.13. chain-a3-propext**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-propext.pem
description:	Contains a proprietary, non-critical extension
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.14. IMB-chain-a3-BER-enc**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-BER-enc.pem
description:	Encoded as BER (not DER)
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Certificate uses BER encoding.

**A.5.3.15. chain-a3-bad-version**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-bad-version.pem
description:	Version field other than X.509v3
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Certificate version is X.509v2.

**A.5.3.16. chain-a3-no-saf**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-saf.pem
description:	Missing SignatureAlgorithm field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	SignatureAlgorithm field is not present

**A.5.3.17. chain-a3-no-svf**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-svf.pem
description:	Missing SignatureValue field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing SignatureValue field

**A.5.3.18. chain-a3-no-ver**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-ver.pem
description:	Missing Version field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing Version field

**A.5.3.19. chain-a3-no-sn**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-sn.pem
description:	Missing SerialNumber field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing SerialNumber field

**A.5.3.20. chain-a3-no-sig**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-sig.pem
description:	Missing Signature field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing Signature field

**A.5.3.21. chain-a3-no-issuer**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-issuer.pem
description:	Missing Issuer field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The Issuer field is not present.

**A.5.3.22. chain-a3-no-subject**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-subject.pem
description:	Missing Subject field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing Subject field

**A.5.3.23. chain-a3-no-spki**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-spki.pem
description:	Missing SubjectPublicKeyInfo field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing SubjectPublicKeyInfo field

**A.5.3.24. chain-a3-no-val-f**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-val-f.pem
description:	Missing Validity field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing Validity field

**A.5.3.25. chain-a3-no-aki-f**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-aki-f.pem
description:	Missing AuthorityKeyIdentifier field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The AuthorityKeyIdentifier is not present.



**A.5.3.26. chain-a3-no-keyuse**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-keyuse.pem
description:	Missing KeyUsage field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The Key Usage field is not present.

**A.5.3.27. chain-a3-no-basic**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-no-basic.pem
description:	Missing BasicConstraint field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The Basic Constraints field is not present.

**A.5.3.28. chain-a3-path-1**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-path-1.pem
description:	Cert.Auth. true, PathLenpresent and zero
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.29. chain-a3-path-2**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-path-2.pem
description:	Cert.Auth. true, PathLen present and positive
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.30. chain-a3-path-3**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-path-3.pem
description:	Cert.Auth. true, PathLen present and negative
conforms to:	S430-2-2006
prerequisites:	None
malformations:	PathLen is -1.

**A.5.3.31. chain-a3-path-4**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-path-4.pem
description:	Cert.Auth. false, PathLen absent
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.32. chain-a3-path-5**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-path-5.pem
description:	Cert.Auth. false, PathLen zero
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.33. chain-a3-path-6**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-path-6.pem
description:	Cert.Auth. false, PathLen positive
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.34. chain-a3-path-7**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-path-7.pem
description:	Cert.Auth. false, PathLen negative
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.3.35. chain-a3-org-name**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-org-name.pem
description:	OrganizationName in subject and issuer fields does not match
conforms to:	S430-2-2006
prerequisites:	None
malformations:	OrganizationName in subject field has first two letters transposed.

**A.5.3.36. chain-a3-role-1**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-role-1.pem
description:	Cert.Auth. False, no role specified in CommonName
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Common Name begins with a period (".")

**A.5.3.37. chain-a3-role-2**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-role-2.pem
description:	Non-SMS role in CN
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Common Name does not include SMS in the section to the left of the first period (".").

**A.5.3.38. chain-a3-date-exp**

Item Data	Data Description
type:	IMB Certificate
filename:	IMB-chain-a3-date-exp.pem
description:	Expired
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Certificate Not After field contains a date value in the past.

**A.5.4. Chain B1 Certificate Files****A.5.4.1. chain-b1-root**

Item Data	Data Description
type:	PRJ Certificate
filename:	PRJ-chain-b1-root.pem
description:	Self-signed root certificate for PRJ devices
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.4.2. chain-b1-signer1**

Item Data	Data Description
type:	PRJ Certificate
filename:	PRJ-chain-b1-signer1.pem
description:	Intermediate Signer, level one
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.4.3. chain-b1-signer2**

Item Data	Data Description
type:	PRJ Certificate
filename:	PRJ-chain-b1-signer2.pem
description:	Intermediate Signer, level two
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

## A.5.5. Chain C1 Certificate Files

### A.5.5.1. chain-c3-root

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-root.pem
description:	Self-signed root certificate for KDS devices
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

### A.5.5.2. chain-c1-signer1

Item Data	Data Description
type:	IMB Certificate
filename:	KDS-chain-c1-signer1.pem
description:	Intermediate Signer, level one
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

### A.5.5.3. KDS-chain-c1-device1

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c1-device1.pem
description:	Normal device certificate
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

## A.5.6. Chain C3 Certificate Files

### A.5.6.1. chain-c1-root

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c1-root.pem
description:	Self-signed root certificate for KDS devices
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.2. chain-c3-signer1**

Item Data	Data Description
type:	IMB Certificate
filename:	KDS-chain-c3-signer1.pem
description:	Intermediate Signer, level one
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.3. chain-a3-osig-type**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-osig-type.pem
description:	Signature algorithm of outside signature is sha1WithRSAEncryption
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.4. chain-a3-isig-type**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-isig-type.pem
description:	Signature algorithm inside signature not sha256WithRSAEncryption
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Signature algorithm inside the signature is sha1WithRSAEncryption

**A.5.6.5. chain-a3-iosig-type**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-iosig-type.pem
description:	Signature algorithm inside and outside identical, but not sha256WithRSAEncryption
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Signature algorithm is sha1 WithRSAEncryption

**A.5.6.6. chain-a3-no-rsa**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-rsa.pem
description:	Public Key not an RSA Key
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Public Key is a DSA key.

**A.5.6.7. chain-a3-short-rsa**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-short-rsa.pem
description:	Public Key Length 1024 bit
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Public key is 1024 bits.

**A.5.6.8. chain-a3-bad-exp**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-bad-exp.pem
description:	Public Key Exponent other than default 65537
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Public Key Exponent is 43.

**A.5.6.9. chain-a3-bad-dnq**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-bad-dnq.pem
description:	dnQualifier in SubjectName field does not match RSA thumbprint
conforms to:	S430-2-2006
prerequisites:	None
malformations:	DnQualifier in Subject field is different than the calculated DnQualifier of the public key.

**A.5.6.10. chain-a3-bad-sig**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-bad-sig.pem
description:	Invalid signature
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Plaintext message digest of signature is different than the message digest of the certificate.

**A.5.6.11. chain-a3-date-ext**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-date-ext.pem
description:	Validity date of child certificate extends beyond parent certificate
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The Not After value of the leaf certificate extends 72 hours past the Not After value of the signer certificate.

**A.5.6.12. chain-a3-propext-crit**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-propext-crit.pem
description:	Contains a proprietary, critical extension
conforms to:	S430-2-2006
prerequisites:	None
malformations:	AltSubjectName is present and marked critical.

**A.5.6.13. chain-a3-propext**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-propext.pem
description:	Contains a proprietary, non-critical extension
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None



**A.5.6.14. KDS-chain-c3-BER-enc**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-BER-enc.pem
description:	Encoded as BER (not DER)
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Certificate uses BER encoding.

**A.5.6.15. chain-a3-bad-version**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-bad-version.pem
description:	Version field other than X.509v3
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Certificate version is X.509v2.

**A.5.6.16. chain-a3-no-saf**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-saf.pem
description:	Missing SignatureAlgorithm field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	SignatureAlgorithm field is not present

**A.5.6.17. chain-a3-no-svf**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-svf.pem
description:	Missing SignatureValue field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing SignatureValue field

**A.5.6.18. chain-a3-no-ver**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-ver.pem
description:	Missing Version field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing Version field

**A.5.6.19. chain-a3-no-sn**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-sn.pem
description:	Missing SerialNumber field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing SerialNumber field

**A.5.6.20. chain-a3-no-sig**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-sig.pem
description:	Missing Signature field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing Signature field

**A.5.6.21. chain-a3-no-issuer**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-issuer.pem
description:	Missing Issuer field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The Issuer field is not present.

**A.5.6.22. chain-a3-no-subject**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-subject.pem
description:	Missing Subject field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing Subject field

**A.5.6.23. chain-a3-no-spki**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-spki.pem
description:	Missing SubjectPublicKeyInfo field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing SubjectPublicKeyInfo field

**A.5.6.24. chain-a3-no-val-f**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-val-f.pem
description:	Missing Validity field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Missing Validity field

**A.5.6.25. chain-a3-no-aki-f**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-aki-f.pem
description:	Missing AuthorityKeyIdentifier field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The AuthorityKeyIdentifier is not present.

**A.5.6.26. chain-a3-no-keyuse**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-keyuse.pem
description:	Missing KeyUsage field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The Key Usage field is not present.

**A.5.6.27. chain-a3-no-basic**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-no-basic.pem
description:	Missing BasicConstraint field
conforms to:	S430-2-2006
prerequisites:	None
malformations:	The Basic Constraints field is not present.

**A.5.6.28. chain-a3-path-1**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-path-1.pem
description:	Cert.Auth. true, PathLenpresent and zero
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.29. chain-a3-path-2**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-path-2.pem
description:	Cert.Auth. true, PathLen present and positive
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.30. chain-a3-path-3**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-path-3.pem
description:	Cert.Auth. true, PathLen present and negative
conforms to:	S430-2-2006
prerequisites:	None
malformations:	PathLen is -1.

**A.5.6.31. chain-a3-path-4**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-path-4.pem
description:	Cert.Auth. false, PathLen absent
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.32. chain-a3-path-5**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-path-5.pem
description:	Cert.Auth. false, PathLen zero
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.33. chain-a3-path-6**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-path-6.pem
description:	Cert.Auth. false, PathLen positive
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.34. chain-a3-path-7**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-path-7.pem
description:	Cert.Auth. false, PathLen negative
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

**A.5.6.35. chain-a3-org-name**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-org-name.pem
description:	OrganizationName in subject and issuer fields does not match
conforms to:	S430-2-2006
prerequisites:	None
malformations:	OrganizationName in subject field has first two letters transposed.

**A.5.6.36. chain-a3-role-1**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-role-1.pem
description:	Cert.Auth. False, no role specified in CommonName
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Common Name begins with a period (".")

**A.5.6.37. chain-a3-date-exp**

Item Data	Data Description
type:	KDS Certificate
filename:	KDS-chain-c3-date-exp.pem
description:	Expired
conforms to:	S430-2-2006
prerequisites:	None
malformations:	Certificate Not After field contains a date value in the past.

**A.5.6.38. KDS chain-a3-role-2**

<b>Item Data</b>	<b>Data Description</b>
type:	KDS Certificate
filename:	KDS-chain-c3-role-2.pem
description:	Cert.Auth. False, role error (TBA).
conforms to:	S430-2-2006
prerequisites:	None
malformations:	None

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

## A.6. Key Delivery Messages

### A.6.1. Introduction

The KDM files defined in this section must be generated for the device under test and the time and date of the test procedure.

### A.6.2. KDM with invalid XML

Item Data	Data Description
type:	KDM
filename:	kdm-malf-any.kdm.xml
description:	KDM that contains an invalid XML file format
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	Missing </DCinemaSecurityMessage> tag

### A.6.3. KDM that has expired

Item Data	Data Description
type:	KDM
filename:	kdm-expired.kdm.xml
description:	KDM that has a validity period that has expired
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The value of the ContentKeysNotValidAfter element is a UTC timestamp at least 24 hours in the past.

### A.6.4. KDM with incorrect message digest

Item Data	Data Description
type:	KDM
filename:	kdm-malf-sig-digest.kdm.xml
description:	KDM in which a Signature Digest has been altered
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The plaintext form of the encrypted message digest in the signature is not the same value as a calculated message digest of the KDM.



### A.6.5. KDM with future validity period

Item Data	Data Description
type:	KDM
filename:	kdm-future.kdm.xml
description:	KDM that has a validity period that is in the future
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The value of the ContentKeysNotValidBefore element is a UTC timestamp at least 24 hours in the future.

### A.6.6. KDM with empty TDL

Item Data	Data Description
type:	KDM
filename:	kdm-no-tdl.kdm.xml
description:	KDM that has an empty Trusted Device List (TDL)
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The DeviceList element of the KDM is empty.

### A.6.7. KDM with imminent expiration date

Item Data	Data Description
type:	KDM
filename:	kdm-short-expire.kdm.xml
description:	KDM that has a validity period that is current, but expires in the near future
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The value of the ContentKeysNotValidAfter element is a UTC timestamp no greater than 60 minutes in the future.

### A.6.8. KDM with no Forensic Marking enabled

Item Data	Data Description
type:	KDM
filename:	kdm-no-fm.kdm.xml
description:	KDM that commands Image Forensic Marking (FM) disabled, and Audio FM disabled
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	A KDM with a ForensicMarkFlag set to <a href="http://www.smp-te-ra.org/430-1/2006/KDM#mrkflg-picture-disable">http://www.smp-te-ra.org/430-1/2006/KDM#mrkflg-picture-disable</a> and <a href="http://www.smp-te-ra.org/430-1/2006/KDM#mrkflg-audio-disable">http://www.smp-te-ra.org/430-1/2006/KDM#mrkflg-audio-disable</a>

### A.6.9. KDM with Image Forensic Marking enabled

Item Data	Data Description
type:	KDM
filename:	kdm-image-only-fm.kdm.xml
description:	KDM that commands Image Forensic Marking (FM) enabled, and Audio FM disabled
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	A KDM with a ForensicMarkFlag set to <a href="http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-audio-disable">http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-audio-disable</a>

### A.6.10. KDM with Audio Forensic Marking enabled

Item Data	Data Description
type:	KDM
filename:	kdm-sound-only-fm.kdm.xml
description:	KDM that commands Image Forensic Marking (FM) disabled, and Audio FM enabled
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	A KDM with a ForensicMarkFlag set to <a href="http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-picture-disable">http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-picture-disable</a> .

### A.6.11. KDM with corrupted CipherData block

Item Data	Data Description
type:	KDM
filename:	kdm-malf-CipherData-block.kdm.xml
description:	KDM that contains an Invalid Structure ID field in the CipherData element
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The first byte of the Structure ID field contained in the <enc:CipherValue> element inside the <enc:CipherData> element has been changed from "F1" to "1F"

**A.6.12. KDM with incorrect signer thumbprint**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-signer-tp.kdm.xml
description:	KDM for which the Thumbprint of the Signer's Certificate does not match the Signer of the KDM
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The thumbprint of the signer certificate as listed in the KDM is incorrect and does not match the thumbprint for the issuing certificate.

**A.6.13. KDM without signer certificate**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-chain.kdm.xml
description:	KDM in which the Certificate chain does not contain the Signer's Certificate
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The certificate that signed the KDM is not included in the KDM.

**A.6.14. KDM without AuthorityKey certificate**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-chain-no-cert-authkeyid.kdm.xml
description:	KDM in which the Certificate chain does not contain the certificate specified by the AuthorityKeyIdentifier value in the Signer Certificate
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	None
malformations:	A KDM that specifies the signer's issuer certificate as the AuthorityKeyIdentifier but which does not contain that certificate.

**A.6.15. KDM with KeyInfo mismatch**

Item Data	Data Description
type:	KDM
filename:	kdm-bad-keyinfo.kdm.xml
description:	KeyInfo field of the audio EncryptedKey element does not match the KeyInfo field of the image EncryptedKey element
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The keyinfo element of the audio encrypted key data contains the (incorrect) Issuer Name and IssuerSerial of the issuer certificate of the KDM signer, the keyinfo element of the image encrypted key contains the (correct) Issuer Name and IssuerSerial of the issuing certificate.

**A.6.16. KDM with mismatched CipherData CPL ID**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-CipherData-cplid.kdm.xml
description:	KDM in which the CompositionPlaylistId in the CipherData element has been altered
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	CompositionPlaylistId field in the CipherData element is different than the CompositionPlaylistId in the AuthenticatedPublic area of the KDM.

**A.6.17. KDM without MessageType**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-missing-MessageType.kdm.xml
description:	KDM with missing MessageType element in XML structure
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	A KDM in which the MessageType element is not present.

**A.6.18. KDM with invalid MessageType**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-bad-MessageType.kdm.xml
description:	KDM with an Invalid MessageType element
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	MessageType element contains: "http://www.smpte-qa.org/430-1/2006/KDM#kdm-key-type"

**A.6.19. KDM with expired Signer certificate**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-expired-signer.kdm.xml
description:	KDM with an expired Signer's Certificate and an ETM IssueDate later than Signer's Certificate expiry date
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	KDM signer's certificate's Validity "Not After" date is earlier than ETM IssueDate

**A.6.20. KDM issued before certificate valid**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-issue-before-cert-valid.kdm.xml
description:	KDM with a valid Signer's Certificate, but ETM issue date before Signer's Certificate issue date
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The <IssueDate> element contains a date prior to the date of the signer certificate's Validity "Not Before" date.

**A.6.21. KDM validity exceeds signer validity**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-signer-cert-exp-before-kdm-expires.kdm.xml
description:	KDM with a validity period that extends beyond the validity of Signer's Certificate expiry date.
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The KDM has a ContentKeysNotValidAfter value later than the signer certificate's "Not After" value.

**A.6.22. KDM without CarriedKeyName**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-no-CarriedKeyName.kdm.xml
description:	EncryptedData field that has no CarriedKeyName element
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

**A.6.23. KDM without EncryptedData**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-no-EncryptedData.kdm.xml
description:	CarriedKeyName field without EncryptedData field
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

**A.6.24. KDM with invalid message digest**

Item Data	Data Description
type:	KDM
filename:	kdm-malf-sig-digest-2.kdm.xml
description:	Invalid message digest (i.e. not matching the signed digest values)
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	The plaintext form of the encrypted message digest in the signature is not the same value as a calculated message digest of the KDM.

### A.6.25. KDM with mismatched keytype

Item Data	Data Description
type:	KDM
filename:	kdm-malf-mismatched-key-keytype.kdm.xml
description:	KDM with an encryption key that is valid but has an incorrect keytype
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	Key is a valid image encryption key but has the keytype "MDAK".

### A.6.26. KDM for multiple LDs, 2 LDBs

Item Data	Data Description
type:	KDM
filename:	kdm-mult-ld-proj1-proj2.kdm.xml
description:	KDM that has a TDL containing thumbprints for the following LDB and Projector SPB type 2 certificates: 1. chain-b1-ldb-1.pem 2. chain-b1-spb2-1.pem 3. chain-b1-ldb-2.pem 4. chain-b1-spb2-2.pem
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

### A.6.27. KDM for multiple LDs, 1 LD/LE, 1 LDB

Item Data	Data Description
type:	KDM
filename:	kdm-mult-ld-proc1-proj1.kdm.xml
description:	KDM that has a TDL containing thumbprints for the following LD/LE, LDB and Projector SPB type 2 certificates: 1. chain-b1-ldle-1.pem 2. chain-b1-ldb-1.pem 3. chain-b1-spb2-1.pem
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

**A.6.28. KDM for multiple LDs, 2 LD/LE, 2 LDB**

Item Data	Data Description
type:	KDM
filename:	kdm-mult-ld-proc1-proj1-proc2-proj2.kdm.xml
description:	KDM that has a TDL containing thumbprints for the following LD/LE, LDB and Projector SPB type 2 certificates: 1. chain-b1-ldle-1.pem 2. chain-b1-ldb-1.pem 3. chain-b1-spb2-1.pem 4. chain-b1-ldle-2.pem 5. chain-b1-ldb-2.pem 6. chain-b1-spb2-2.pem
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

**A.6.29. KDM for multiple LDs, 2 LD/LE, 1 LDB**

Item Data	Data Description
type:	KDM
filename:	kdm-mult-ld-proc1-proc2-proj1.kdm.xml
description:	KDM that has a TDL containing thumbprints for the following LD/LE, LDB and Projector SPB type 2 certificates: 1. chain-b1-ldle-1.pem 2. chain-b1-ldb-1.pem 3. chain-b1-spb2-1.pem 4. chain-b1-ldle-2.pem
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

**A.6.30. KDM for 2K StEM**

Item Data	Data Description
type:	KDM
filename:	2K_StEM_sequence_ct-kdm.kdm.xml
description:	
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None



**A.6.31. Expired KDM for 2K StEM**

Item Data	Data Description
type:	KDM
filename:	2K_StEM_sequence_ct-kdm-expired.kdm.xml
description:	
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

**A.6.32. Image FM only KDM for 2K StEM**

Item Data	Data Description
type:	KDM
filename:	2K_StEM_sequence_ct-kdm-image-only-fm.kdm.xml
description:	
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

**A.6.33. No FM KDM for 2K StEM**

Item Data	Data Description
type:	KDM
filename:	2K_StEM_sequence_ct-kdm-no-fm.kdm.xml
description:	
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

**A.6.34. Sound Only FM KDM for 2K StEM**

Item Data	Data Description
type:	KDM
filename:	2K_StEM_sequence_ct-kdm-sound-only-fm.kdm.xml
description:	
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

### A.6.35. KDM for 128 Reel Composition, "A" Series

Item Data	Data Description
type:	KDM
filename:	2K_StEM_128_a_reels_ct-kdm.kdm.xml
description:	
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_128_a_reels_ct.cpl
malformations:	None

### A.6.36. KDM for 128 Reel Composition, "B" Series

Item Data	Data Description
type:	KDM
filename:	2K_StEM_128_b_reels_ct-kdm.kdm.xml
description:	
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_128_b_reels_ct.cpl
malformations:	None

### A.6.37. FM Constraints

Item Data	Data Description
type:	KDM
filename:	2K_fm_constraints_ct-kdm.kdm.xml
description:	
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_pt.cpl, 2K_StEM_sequence_ct.cpl
malformations:	None

### A.6.38. KDM with non-empty NonCriticalExtensions

Item Data	Data Description
type:	KDM
filename:	kdm-with-non-crit-exts.kdm.xml
description:	KDM with a non-empty NonCriticalExtensions element
conforms to:	S430-1-2006, S430-3-2006
prerequisites:	2K_StEM_sequence_ct.cpl
malformations:	None

# Appendix B. Equipment List

## B.1. Hardware

AES3 Audio Analyzer	Digital audio signal analyzer with AES-3 inputs.
Sound System	5.1 or 7.1 sound system with calibrated level control.
Computer with POSIX OS	Computer with POSIX-like Operating System (OS), such as Linux or Mac OS X. The system must support TCP/IP via 1000 Mb/s ethernet and be backward-compatible to support 100 Mb/s Ethernet.
Digital Clock	Digital quartz time-of-day clock displaying time accurate to the second
Oscilloscope	Digital storage oscilloscope, 200 MHz or better, with two or more inputs plus external trigger.
Accurate Real-Time Clock	A real-time clock that uses an external reference to maintain precise time (within 1 ms). The external reference should be WWV, GPS or NTP traceable to a trusted hardware clock.
DMM	Digital Multimeter, A.C. / D.C. operation, measures Volts, Amperes and Ohms to at least 3 decimal places.
FM Decoder	Forensic mark decoder for image or sound essence. The exact type of decoder is dependent upon the type of watermark to be decoded. This equipment is expected to collect the full payload of the forensic marking system.
FM Detector	Forensic mark detector for image or sound essence. The exact type of detector is dependent upon the type of watermark to be detected. This equipment is expected to simply detect the presence of the forensic mark.
Ethernet Switch	A 1000Base-T Ethernet switch capable of sustained full-rate throughput on at least two port-pairs. The device must also be able to configure one or more ports as "monitor" ports (selected traffic on the switch can be copied to the monitor port to facilitate diagnostic capture).
Photodiode	Photodiode, of the type most sensitive in the human visible electromagnetic spectrum (about 390 nm to 780 nm), fitted with suitable length of shielded cable, terminated in a BNC connector.
Photometer	Photometer as described in [SMPTE-431-1].
Spectroradiometer	Spectroradiometer as described in [SMPTE-431-1].
Stopwatch	Digital stopwatch with .01 second resolution.
Still Camera	Digital still image camera.
48 fps Camera	Camera and recorder/reproducer system capable of 48fps (or better) capture rate.
SPB-2 Access Tools	Tools, supplied by the manufacturer of an SPB-2 device, required to gain authorized access to the protected area of the SPB-2.
D.U.T. Twin	A device identical to the Device Under Test.
DCI Projector Pair	Two DCI-complaint projectors of the same model and revision.

Dual-Link Monitor	Dual-link HD monitor.
Bridge Tap Connector	Connector with a bridge tap takeoff point, e.g. a "BNC Tee".
GPIO Test Fixture	A test fixture comprising L.E.D. indicators, switches and a power supply, per the schematic in Appendix E.
DCI Projector	DCI-compliant standalone projector.
DCI Server	DCI-compliant server (includes Image/Sound Media Block)

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

## B.2. Software

Audio Editor	A digital audio workstation (DAW), such as Pro Tools or Audacity.
asm-responder	Auditorium Security Message (ASM) responder simulator. Allows inspection of ASM communications behavior in a peer device. See Appendix D for more information.
asm-requester	Auditorium Security Message (ASM) requester simulator. Allows inspection of ASM communications behavior in a peer device. See Appendix D for more information.
ftlint	The ftlint command line utility from the FreeType library. It is available from <a href="http://www.freetype.org/">http://www.freetype.org/</a> .
Network Analyzer	Network analysis tool such as Wireshark or <b>tcpdump</b> . It is available from <a href="http://www.wireshark.org/">http://www.wireshark.org/</a> .
Text Editor	Any text editor that can display and write plain text, such as <b>emacs</b> or <b>vi</b> , etc.
OpenJPEG	JPEG 2000 encoder/decoder software such as OpenJPEG. It is available from <a href="http://www.openjpeg.org/">http://www.openjpeg.org/</a> .
j2c-scan	JPEG 2000 scanner based on the OpenJPEG library. The source code for this program is available in Section C.8.
identify	<b>identify</b> is part of the ImageMagick library and utility suite. It is available from <a href="http://www.imagemagick.org/">http://www.imagemagick.org/</a> .
klvwalk	The <b>klvwalk</b> utility from the free asdcplib software package. It is available from <a href="http://www.cinecert.com/asdcplib/">http://www.cinecert.com/asdcplib/</a> .
asdcplib-test	The <b>asdcplib-test</b> utility from the free asdcplib software package. It is available from <a href="http://www.cinecert.com/asdcplib/">http://www.cinecert.com/asdcplib/</a> .
openssl	General purpose command line utility from the OpenSSL software package. It is available from <a href="http://www.openssl.org/">http://www.openssl.org/</a> .
schema-check	Validating XML Parser (note: may use parser from Xerces package). The source code for this program is available in Section C.3. It is available from <a href="http://xml.apache.org/security/">http://xml.apache.org/security/</a> .
checksig	XML Signature validator, distributed with the XML Security library. It is available from <a href="http://xml.apache.org/security/">http://xml.apache.org/security/</a> .
dsig_cert.py	XML Signature certificate manipulator. The source code for this program is available in Section C.8.

dsig_extract.py	XML Signature certificate extractor. The source code for this program is available in Section C.9.
uuid_check.py	UUID validator. The source code for this program is available in Section C.7.
dc-thumbprint	Certificate thumbprint calculator. The source code for this program is available in Section C.2.
eab_calc.py	Delta E*ab Calculator for color accuracy measurements. The source code for this program is available in Section C.6.
kdm-decrypt	KDM decryption tool. The source code for this program is available in Section C.4.

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

---

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

**Page Intentionally Left Blank**

---

# Appendix C. Source Code

## C.1. Overview

Wherever possible, the computer programs used in the test procedures in this document are freely available. Where appropriate, the listings in Appendix B provide a URL where the software can be obtained.

In some cases, it was necessary to develop programs because free alternatives were not available. Those programs are presented here in source code form along with instructions for building and executing the programs.

The programs are expressed in the C, C++ and Python programming languages. Build instructions and prerequisites for the C and C++ programs are given in the comments at the beginning of each source module. Machine readable copies of the programs are available in the `source-code` directory in the Reference Materials distribution (see Appendix A).

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

## C.2. dc-thumbprint

This program reads a PEM formatted X509 certificate and calculates a SHA-1 message digest over the signed portion of the certificate as required by [SMPTE 430-2-2006]. The value is encoded as a Base64 string and returned on stdout. The following example illustrates this usage:

### Example C.1. dc-thumbprint execution

```
$ dc-thumbprint my-cert.pem
aZMVnZ/TzEvLUCmQFcc8U0je9uo=
```

### C.2.1. dc-thumbprint Source Code Listing

```
/*
 * dc-thumbprint.c -- calculate certificate thumbprint of PEM-encoded
 *                   X.509 document per SMPTE 430-2
 *
 *
 * This program requires OpenSSL. To build:
 * $ cc -o dc-thumbprint dc-thumbprint.c -lcrypto
 */

#include <stdio.h>
#include <string.h>
#include <openssl/sha.h>
#include <openssl/pem.h>

typedef unsigned char byte_t;

char*
encodeBase64(byte_t* in_buf, int in_len, char* out_buf, int out_len)
{
    BIO *bmem, *b64;
    BUF_MEM *bptr;

    b64 = BIO_new(BIO_f_base64());
    bmem = BIO_new(BIO_s_mem());
    b64 = BIO_push(b64, bmem);
    BIO_write(b64, in_buf, in_len);
    BIO_flush(b64);
    BIO_get_mem_ptr(b64, &bptr);

    if ( bptr->length + 1 > out_len )
    {
        fprintf(stderr, "\n");
        return 0;
    }

    memcpy((byte_t*)out_buf, bptr->data, bptr->length-1);
    out_buf[bptr->length-1] = 0;

    return out_buf;
}

int
main(int argc, char** argv)
{
    byte_t  sha_value[20];    /* buffer for resulting thumbprint digest */
    char    sha_base64[64]; /* buffer for Base64 version of the thumbprint digest */
```



```

byte_t  p_key_buf[8192]; /* buffer holds DER encoded certificate body */
size_t  length;        /* length of DER encoded certificate body (p_key_buf) */
byte_t* p = p_key_buf; /* pointer that OpenSSL will move at will */
int     i;             /* character iterator for output stage */
SHA_CTX SHA;          /* SHA-1 context for thumbprint */
FILE*   fp;           /* PEM source file */
X509*   x509obj;      /* X509 object for mangling certificate contents */

OpenSSL_add_all_digests();

if ( argc != 2 )
{
    fprintf(stderr, "USAGE: dc-thumbprint cert-file.pem\n");
    return 1;
}

if ( (fp = fopen (argv[1], "r")) == 0 )
{
    perror("fopen");
    return 2;
}

if ( (x509obj = PEM_read_X509(fp, 0, 0, 0)) == 0 )
{
    fprintf(stderr, "Error decoding file %s\n", argv[1]);
    fclose (fp);
    return 3;
}

fclose (fp);

/* get the certificate body as a DER string */
if ( i2d_X509_CINF(x509obj->cert_info, &p) == 0 )
{
    fprintf(stderr, "i2d_X509_CINF error\n");
    return 4;
}

length = p - p_key_buf;

if ( length > 8192 )
{
    fprintf(stderr, "i2d_X509_CINF value exceeds buffer length\n");
    return 5;
}

SHA1_Init(&SHA);
SHA1_Update(&SHA, p_key_buf, length);
SHA1_Final(sha_value, &SHA);

if ( encodeBase64(sha_value, 20, sha_base64, 64) == 0 )
    return 6;

printf("%s\n", sha_base64);
return 0;
}

/*
 * end dc-thumbprint.c
 */

```

## C.3. schema-check

This program parses and validates XML instance documents. When an XML document is specified alone, the file is checked for well-formedness but is not validated. When an XML document is specified with one or more schema files, **schema-check** validates that file against the schemas. Only one file to be tested may be specified at a time. The following example illustrates using the program to check well-formedness:

### Example C.2. Using schema-check to check well-formedness

```
$ schema-check perfect-movie.cpl.xml
```

The next example shows how to use the program to check for valid content:

### Example C.3. Using schema-check to check validity

```
$ schema-check perfect-movie.cpl.xml SMPTE-428-7-2007.xsd
```

### C.3.1. schema-check Source Code Listing

```
//
// schema-check.cpp -- test XML document against schema
//
//
// This program requires the Xerces-c XML library. To build:
// $ c++ -o schema-check schema-check.cpp -lxerces-c
//

#include <iostream>
#include <list>
#include <string>

#include <xercesc/util/OutOfMemoryException.hpp>
#include <xercesc/dom/DOM.hpp>
#include <xercesc/parsers/XercesDOMParser.hpp>
#include <xercesc/framework/XMLGrammarDescription.hpp>
#include <xercesc/sax/ErrorHandler.hpp>
#include <xercesc/sax/SAXParseException.hpp>

using std::cerr;
using std::endl;
XERCES_CPP_NAMESPACE_USE

// -----
// Utility code adapted from the DOMPrint program distributed with Xerces-c

// simple transcoding wrapper
class StrX
{
    char* fLocalForm;

public:
    StrX(const XMLCh* const toTranscode) { fLocalForm = XMLString::transcode(toTranscode); }
    ~StrX() { XMLString::release(&fLocalForm); }
    const char* localForm() const { return fLocalForm; }
};

std::ostream&
operator<<(std::ostream& target, const StrX& toDump)
```

```

{
    target << toDump.localForm();
    return target;
}

// error handler interface
class DOMTreeErrorReporter : public ErrorHandler
{
public:
    void warning(const SAXParseException& toCatch) {}
    void resetErrors() {}

    void DOMTreeErrorReporter::error(const SAXParseException& toCatch) {
        cerr << "Error at file \"" << StrX(toCatch.getSystemId())
            << "\", line " << toCatch.getLineNumber()
            << ", column " << toCatch.getColumnNumber() << endl
            << "    Message: " << StrX(toCatch.getMessage()) << endl;
    }

    void DOMTreeErrorReporter::fatalError(const SAXParseException& toCatch) {
        cerr << "Fatal Error at file \"" << StrX(toCatch.getSystemId())
            << "\", line " << toCatch.getLineNumber()
            << ", column " << toCatch.getColumnNumber() << endl
            << "    Message: " << StrX(toCatch.getMessage()) << endl;
    }
};

// -----

int
main(int argc, const char** argv)
{
    try
    {
        XMLPlatformUtils::Initialize();
    }
    catch(const XMLException& e)
    {
        StrX tmp_e(e.getMessage());
        cerr << "Xerces initialization error: " << tmp_e.localForm() << endl;
        return 2;
    }

    // check command line for arguments
    if ( argc < 1 )
    {
        cerr << "usage: schema-check <xml-file> [<schema-file> ...]" << endl;
        return 3;
    }

    XercesDOMParser *parser = new XercesDOMParser;
    DOMTreeErrorReporter *errReporter = new DOMTreeErrorReporter();
    parser->setErrorHandler(errReporter);

    parser->setDoNamespaces(true);
    parser->setCreateEntityReferenceNodes(true);
    parser->useCachedGrammarInParse(true);

    if ( argc > 2 )
    {
        parser->setDoSchema(true);
        parser->setDoValidation(true);
        parser->setValidationSchemaFullChecking(true);

        for ( int i = 2; i < argc; i++ )
        {
            if ( parser->loadGrammar(argv[i], Grammar::SchemaGrammarType, true) == 0 )

```

```
        {
            cerr << "Error loading grammar " << std::string(argv[i]) << endl;
            return 4;
        }
    }
}

bool errorsOccured = true;
try
{
    parser->parse(argv[1]);
    errorsOccured = false;
}
catch ( const OutOfMemoryException& )
{
    cerr << "Out of memory exception." << endl;
}
catch ( const XMLException& e )
{
    cerr << "An error occurred during parsing" << endl
        << "    Message: " << StrX(e.getMessage()) << endl;
}
catch ( const DOMException& e )
{
    const unsigned int maxChars = 2047;
    XMLCh errText[maxChars + 1];

    cerr << endl
        << "DOM Error during parsing: '" << std::string(argv[1]) << "'" << endl
        << "DOM Exception code is: " << e.code << endl;

    if ( DOMImplementation::loadDOMExceptionMsg(e.code, errText, maxChars) )
        cerr << "Message is: " << StrX(errText) << endl;
}
catch (...)
{
    cerr << "An error occurred during parsing." << endl;
}

return errorsOccured ? 1 : 0;
}

//
// end schema-check.cpp
//
```

## C.4. kdm-decrypt

This program reads a KDM and an RSA private key in PEM format and decrypts the EncryptedKey elements in the KDM. The decrypted key blocks are printed to stdout. Note that key blocks in the KDM must have been encrypted using the public key which corresponds to the RSA key given as the second argument to this program.

### Example C.4. kdm-decrypt execution

```
$ kdm-decrypt test_file.kdm.xml my_id_key.pem
CipherDataID: f1dc124460169a0e85bc300642f866ab
SignerThumbprint: q50qr6GkfG6W2HzcBTee5m0Qjzw=
  CPL Id: 119d8990-2e55-4114-80a2-e53f3403118d
  Key Id: b6276c4b-b832-4984-aab6-250c9e4f9138
  Key Type: MDIK
  Not Before: 2007-09-20T03:24:53-00:00
  Not After: 2007-10-20T03:24:53-00:00
  Key Data: 7f2f711f1b4d44b83e1dd1bf90dc7d8c
```

### C.4.1. kdm-decrypt Source Code Listing

```
//
// kdm-decrypt.cpp -- decrypt and display KDM EncryptedKey elements
//
//
// This program requires the Xerces-c XML, XMLSecurity, OpenSSL
// and asdcplib libraries. To build:
//
// c++ -o kdm-decrypt kdm-decrypt.cpp \
//     -lxerces-c -lxml-security-c -lkumu -lcrypto
//
#include <KM_util.h>
#include <KM_fileio.h>
#include <iostream>
#include <string>
#include <openssl/pem.h>
#include <xercesc/util/OutOfMemoryException.hpp>
#include <xercesc/parsers/XercesDOMParser.hpp>
#include <xercesc/framework/MemBufInputSource.hpp>
#include <xsec/framework/XSECPProvider.hpp>
#include <xsec/framework/XSECException.hpp>
#include <xsec/enc/XSECCryptoException.hpp>
#include <xsec/enc/OpenSSL/OpenSSLCryptoKeyRSA.hpp>

XERCES_CPP_NAMESPACE_USE
using std::cout;
using std::cerr;
using std::endl;
using namespace Kumu;

const size_t KeyType_Length = 4;
const size_t DateTime_Length = 25;
const ui32_t X509Thumbprint_Length = 20;

// A structure to hold key block data retrieved during a decrypt operation.
struct S430_2_KeyBlock
{
  byte_t CipherDataID[UUID_Length];
  byte_t SignerThumbprint[X509Thumbprint_Length];
  byte_t CPLId[UUID_Length];
  byte_t KeyType[KeyType_Length];
};
```

```

byte_t KeyId[UUID_Length];
byte_t NotBefore[DateTime_Length];
byte_t NotAfter[DateTime_Length];
byte_t KeyData[SymmetricKey_Length];

S430_2_KeyBlock() {
    memset(this, 0, sizeof(S430_2_KeyBlock));
}

std::string Dump() const;
};

// Pretty-print key block data.
std::string
S430_2_KeyBlock::Dump() const
{
    using std::string;
    Kumu::Identifier<X509Thumbprint_Length> TmpThumbprint;
    UUID    TmpUUID;
    char    tmp_buf[64];
    string  out_string;

    bin2hex(CipherDataID, UUID_Length, tmp_buf, 64);
    out_string = "    CipherDataID: " + string(tmp_buf);
    TmpThumbprint.Set(SignerThumbprint);
    out_string += "\nSignerThumbprint: " + string(TmpThumbprint.EncodeBase64(tmp_buf, 64));
    TmpUUID.Set(CPLId);
    out_string += "\n    CPL Id: " + string(TmpUUID.EncodeHex(tmp_buf, 64));
    TmpUUID.Set(KeyId);
    out_string += "\n    Key Id: " + string(TmpUUID.EncodeHex(tmp_buf, 64));
    out_string += "\n    Key Type: " + KeyType[0] + KeyType[1] + KeyType[2] + KeyType[3];
    assert(DateTime_Length<64);
    tmp_buf[DateTime_Length] = 0;
    memcpy(tmp_buf, NotBefore, DateTime_Length);
    out_string += "\n    Not Before: " + string(tmp_buf);
    memcpy(tmp_buf, NotAfter, DateTime_Length);
    out_string += "\n    Not After: " + string(tmp_buf);
    bin2hex(KeyData, UUID_Length, tmp_buf, 64);
    out_string += "\n    Key Data: " + string(tmp_buf);
    out_string += "\n";
    return out_string;
}

// Given a KDM string and a parsed RSA key, decrypt the key blocks
// in the KDM and print them to stdout.
int
decrypt_kdm(const std::string& KDMDocument, EVP_PKEY* Target)
{
    assert(Target);

    XercesDOMParser* parser = new XercesDOMParser;
    parser->setDoNamespaces(true);
    parser->setCreateEntityReferenceNodes(true);

    try
    {
        MemBufInputSource xmlSource(reinterpret_cast<const XMLByte*>(KDMDocument.c_str()),
                                    static_cast<const unsigned int>(KDMDocument.length()),
                                    "pidc_rules_file");

        parser->parse(xmlSource);
        int errorCount = parser->getErrorCount();
        if ( errorCount > 0 )
        {
            cerr << "XML parse errors: " << errorCount << endl;
            return -1;
        }
    }
}

```

```

catch ( const OutOfMemoryException& )
{
    cerr << "Out of memory exception." << endl;
}
catch ( const XMLException& e )
{
    char* emsg = XMLString::transcode(e.getMessage());
    cerr << "An error occurred during parsing" << endl
         << "    Message: " << emsg << endl;
    XSEC_RELEASE_XMLCH(emsg);
}
catch ( const DOMException& e )
{
    const unsigned int maxChars = 2047;
    XMLCh errText[maxChars + 1];

    cerr << endl
         << "DOM Exception code is: " << e.code << endl;

    if ( DOMImplementation::loadDOMExceptionMsg(e.code, errText, maxChars) )
    {
        char* emsg = XMLString::transcode(errText);
        cerr << "Message is: " << emsg << endl;
        XSEC_RELEASE_XMLCH(emsg);
    }
}
catch (...)
{
    cerr << "Unexpected XML parser error." << endl;
}

try
{
    XSECProvider prov;
    OpenSSLCryptoKeyRSA* PrivateKey = new OpenSSLCryptoKeyRSA(Target);
    if ( PrivateKey == 0 )
    {
        cerr << "Error reading private key" << endl;
        return -1;
    }

    DOMDocument* doc = parser->getDocument();
    assert(doc);
    XENCCipher* cipher = prov.newCipher(doc);
    cipher->setKEK(PrivateKey);

    DOMNodeIterator* Iter =
        ((DOMDocumentTraversal*)doc)->createNodeIterator(doc,
                                                         (DOMNodeFilter::SHOW_ELEMENT),
                                                         0, false);

    assert(Iter);
    DOMNode* Node;
    int keys_accepted = 0;
    int key_nodes_found = 0;

    while ( (Node = Iter->nextNode()) != 0 )
    {
        char* n = XMLString::transcode(Node->getLocalName());
        if ( n == 0 ) continue;

        if ( strcmp(n, "EncryptedKey") == 0 )
        {
            key_nodes_found++;
            S430_2_KeyBlock CipherData;
            ui32_t decrypt_len =
                cipher->decryptKey((DOMElement*)Node,
                                   (byte_t*)&CipherData, sizeof(CipherData));

```

```

        if ( decrypt_len == sizeof(CipherData) )
        {
            keys_accepted++;
            cout << CipherData.Dump();
        }
        else if ( decrypt_len > 0 )
            cerr << "Unexpected cipher block length: " << decrypt_len << endl;
        else
            cerr << "Error decoding key block: " << key_nodes_found << endl;
    }

    XSEC_RELEASE_XMLCH(n);
}

    Iter->release();
}
catch (XSECException &e)
{
    char* emsg = XMLString::transcode(e.getMsg());
    cerr << "Key decryption error: " << emsg << endl;
    XSEC_RELEASE_XMLCH(emsg);
    return -1;
}
catch (XSECCryptoException &e)
{
    cerr << "Crypto error: " << e.getMsg() << endl;
    return -1;
}
catch (...)
{
    cerr << "Unexpected decryption error." << endl;
}

delete parser;
return 0;
}

//
main(int argc, const char** argv)
{
    if ( argc < 3 )
    {
        cerr << "USAGE: kdm-decrypt <kdm-file> <RSA-PEM-file>" << endl;
        return 2;
    }

    try
    {
        XMLPlatformUtils::Initialize();
        XSECPlatformUtils::Initialise();
    }
    catch(const XMLException& e)
    {
        char* emsg = XMLString::transcode(e.getMessage());
        cerr << "Xerces or XMLSecurity initialization error: " << emsg << endl;
        XSEC_RELEASE_XMLCH(emsg);
        return 3;
    }
    catch (...)
    {
        cerr << "Unexpected Xerces or XMLSecurity initialization error." << endl;
    }

    FILE* fp = fopen (argv[2], "r");
    if ( fp == 0 )
    {

```



```
        perror(argv[2]);
        return 4;
    }

    EVP_PKEY* Target = PEM_read_PrivateKey(fp, 0, 0, 0);
    fclose(fp);

    if ( Target == 0 )
    {
        cerr << "Error reading RSA key in file " << std::string(argv[2]) << endl;
        return 5;
    }

    std::string XML_doc;
    Result_t result = ReadFileIntoString(argv[1], XML_doc);
    if ( KM_FAILURE(result) )
    {
        cerr << "Error reading XML file " << std::string(argv[1]) << endl;
        return 6;
    }

    if ( decrypt_kdm(XML_doc, Target) != 0 )
        return 1;

    return 0;
}

//
// end kdm-decrypt.cpp
//
```

ONLY  
FILE  
9

## C.5. j2c-scan

This program reads a JPEG 2000 codestream from a file and produces parametric data on the standard output. The following example illustrates this usage:

### Example C.5. j2c-scan execution

```
$ j2c-scan test_frame_000002.j2c
coding parameters
digital cinema profile: none
rsiz capabilities: standard
pixel offset from top-left corner: (0, 0)
tile width/height in pixels: (2048, 1080)
image width/height in tiles: (1, 1)
tile #1
  coding style: 1
  progression order: Component-Position-Resolution-Layer
  POC marker flag: 0
  number of quality layers: 1
    rate for layer #1: 0.0
  multi-component transform flag: 1
  component #1
    coding style: 1
    number of resolutions: 6
    code block width/height: (5, 5)
    code block coding style: 0
    discrete wavelet transform identifier: 0
    quantization style: 2
    number of guard bits: 1
    step size pairs: 16
    region of interest shift: 0
  component #2
    coding style: 1
    number of resolutions: 6
    code block width/height: (5, 5)
    code block coding style: 0
    discrete wavelet transform identifier: 0
    quantization style: 2
    number of guard bits: 1
    step size pairs: 16
    region of interest shift: 0
  component #3
    coding style: 1
    number of resolutions: 6
    code block width/height: (5, 5)
    code block coding style: 0
    discrete wavelet transform identifier: 0
    quantization style: 2
    number of guard bits: 1
    step size pairs: 16
    region of interest shift: 0
```

### C.5.1. j2c-scan Source Code Listing

```
/*
 * j2c-scan.cpp -- parse j2c file and display data concerning it
 *
 *
 * This program requires the OpenJPEG library. Furthermore, it
 * requires the header files "openjpeg.h" and "j2k.h" from the
 * OpenJPEG source distribution. Copy the headers to your build
```

```

* directory. After doing so, execute the following to build:
* $ c++ -o j2c-scan j2c-scan.cpp -lopenjpeg
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "openjpeg.h"
#include "j2k.h"

static void
j2k_dump_cp (opj_image_t * image, opj_cp_t * cp)
{
    char *s;
    int i, j, k;
    int step_size_pairs;
    printf ("coding parameters\n");
    if (cp->comment != NULL)
    {
        printf (" coding comment: %s\n", cp->comment);
    }
    switch (cp->cinema)
    {
        case OFF:          s = "none";          break;
        case CINEMA2K_24:  s = "2k @ 24 fps";    break;
        case CINEMA2K_48:  s = "2k @ 48 fps";    break;
        case CINEMA4K_24:  s = "4k @ 24 fps";    break;
        default:          s = "unknown";        break;
    }
    printf (" digital cinema profile: %s\n", s);
    switch (cp->rsiz)
    {
        case STD_RSIZ:    s = "standard";        break;
        case CINEMA2K:    s = "2k digital cinema"; break;
        case CINEMA4K:    s = "4k digital cinema"; break;
        default:          s = "unknown";        break;
    }
    printf (" rsiz capabilities: %s\n", s);
    printf (" pixel offset from top-left corner: (%d, %d)\n", cp->tx0,
        cp->ty0);
    printf (" tile width/height in pixels: (%d, %d)\n", cp->tdx, cp->tdy);
    printf (" image width/height in tiles: (%d, %d)\n", cp->tw, cp->th);
    for (i = 0; i < cp->tw * cp->th; i++)
    {
        printf (" tile #%d\n", i + 1);
        printf (" coding style: %x\n", cp->tcps[i].csty);
        switch (cp->tcps[i].prg)
        {
            case LRCP:    s = "Layer-Resolution-Component-Position"; break;
            case RLCP:    s = "Resolution-Layer-Component-Position"; break;
            case RPCL:    s = "Resolution-Position-Component-Layer"; break;
            case PCRL:    s = "Position-Component-Resolution-Layer"; break;
            case CPRL:    s = "Component-Position-Resolution-Layer"; break;
            default:      s = "unknown";          break;
        }
        printf (" progression order: %s\n", s);
        printf (" POC marker flag: %d\n", cp->tcps[i].POC);
        printf (" number of quality layers: %d\n", cp->tcps[i].numlayers);
        for (j = 0; j < cp->tcps[i].numlayers; j++)
        {
            printf (" rate for layer #%d: %.1f\n", j + 1,
                cp->tcps[i].rates[j]);
        }
        printf (" multi-component transform flag: %d\n", cp->tcps[i].mct);
        for (j = 0; j < image->numcomps; j++)
        {
            printf (" component #%d\n", j + 1);
        }
    }
}

```

```

    printf ("      coding style: %x\n", cp->tcps[i].tccps[j].csty);
    printf ("      number of resolutions: %d\n",
            cp->tcps[i].tccps[j].numresolutions);
    printf ("      code block width/height: (%d, %d)\n",
            cp->tcps[i].tccps[j].cblkw, cp->tcps[i].tccps[j].cblkh);
    printf ("      code block coding style: %x\n",
            cp->tcps[i].tccps[j].cblksty);
    printf ("      discrete wavelet transform identifier: %d\n",
            cp->tcps[i].tccps[j].qmfbid);
    printf ("      quantization style: %d\n",
            cp->tcps[i].tccps[j].qntsty);
    printf ("      number of guard bits: %d\n",
            cp->tcps[i].tccps[j].numgbits);
    step_size_pairs =
        (cp->tcps[i].tccps[j].qntsty ==
         J2K_CCP_QNTSTY_SIQNT) ? 1 : cp->tcps[i].tccps[j].numresolutions *
        3 - 2;
    printf ("      step size pairs: %d\n", step_size_pairs);
    printf ("      region of interest shift: %d\n",
            cp->tcps[i].tccps[j].roishift);
    }
}
}

void
error_callback (const char *msg, void *client_data)
{
    FILE *stream = (FILE *) client_data;
    fprintf (stream, "[ERROR] %s", msg);
}

void
warning_callback (const char *msg, void *client_data)
{
    FILE *stream = (FILE *) client_data;
    fprintf (stream, "[WARNING] %s", msg);
}

int
main (int argc, char *argv[])
{
    char *filename;          /* name of the file to process */
    FILE *fp;               /* input file pointer */
    int file_length;        /* length of the input file */
    unsigned char *buffer = NULL; /* in-memory buffer containing the input file */
    opj_cio_t *cio = NULL;  /* OpenJPEG wrapper around file buffer */
    opj_dparameters_t parameters; /* decompression parameters */
    opj_dinfo_t *dinfo = NULL; /* pointer to a JPEG-2000 decompressor */
    opj_event_mgr_t event_mgr; /* manager of events' callback functions */
    opj_image_t *image = NULL; /* pointer to the decoded image */

    memset (&event_mgr, 0, sizeof (opj_event_mgr_t));
    event_mgr.error_handler = error_callback;
    event_mgr.warning_handler = warning_callback;
    event_mgr.info_handler = NULL;

    /* establish default decoding parameters for JPEG-2000 codestreams */
    opj_set_default_decoder_parameters (&parameters);
    parameters.decod_format = 0;

    if (argc != 2)
    {
        fprintf (stderr, "USAGE: j2c-scan file.j2c\n");
        return 1;
    }
    filename = argv[1];

```

```
strncpy (parameters.infile, filename, sizeof (parameters.infile) - 1);

/* read the input file and put it in memory */
fp = fopen (parameters.infile, "rb");
if (fp == NULL)
{
    perror ("fopen");
    return 2;
}
fseek (fp, 0, SEEK_END);
file_length = (int) ftell (fp);
fseek (fp, 0, SEEK_SET);
buffer = (unsigned char *) malloc (file_length);
fread (buffer, sizeof (unsigned char), file_length, fp);
fclose (fp);

/* decode the JPEG-2000 codestream */
dinfo = opj_create_decompress (CODEC_J2K);
opj_set_event_mgr ((opj_common_ptr) dinfo, &event_mgr, stderr);
opj_setup_decoder (dinfo, &parameters);
cio = opj_cio_open ((opj_common_ptr) dinfo, buffer, file_length);
image = opj_decode (dinfo, cio);
if (image == NULL)
{
    fprintf (stderr, "ERROR -> j2c-scan: failed to decode image!\n");
    opj_destroy_decompress (dinfo);
    opj_cio_close (cio);
    free (buffer);
    return 1;
}
opj_cio_close (cio);
free (buffer);

/* display information about the image */
j2k_dump_cp (image, ((opj_j2k_t *) dinfo->j2k_handle)->cp);

/* free the memory */
opj_destroy_decompress (dinfo);
opj_image_destroy (image);

return 0;
}
```

## C.6. Eab\_calc.py

This program reads a measured set of  $xyY$  values and a set of reference values and calculates the  $\Delta E^*_{ab}$  value of the two. This calculation is required to perform the test in Section 7.5.12 The following example illustrates this usage:

### Example C.6. Eab\_calc.py execution

```
$ Eab_calc.py 0.2650 0.6900 34.64 0.2719 0.6835 34.64
L=88.0 a*=-110.2 b*=106.1
L=88.0 a*=-106.2 b*=106.0
DeltaE=4.0
```

### C.6.1. Eab\_calc.py Source Code Listing

```
#!/usr/bin/python
#
# Eab_calc.py -- Calculate Delta E*ab from xyY inputs.
#           Adapted from the examples in SMPTE EG432-1.
#
#
import sys

_Xwhite = 42.940 # d-cinema reference white constants
_Ywhite = 48.0
_Zwhite = 45.81

def _Lab_f1(measured, white_ref):
    q = measured / white_ref
    if q > 0.008856: return pow(q, 1.0/3)
    return 903.3 * q

class Lab_set:
    def init_with_xyY(self, x, y, Y):
        X = ( x / y ) * Y
        z = 1 - x - y
        Z = ( z / y ) * Y
        return self.init_with_XYZ(X,Y,Z)

    def init_with_XYZ(self, X, Y, Z):
        Yratio = _Lab_f1(Y, _Ywhite);
        self.L = 116.0 * Yratio - 16;
        self.a = 500.0 * ( _Lab_f1(X, _Xwhite) - Yratio );
        self.b = 200.0 * ( Yratio - _Lab_f1(Z, _Zwhite) );
        return self

    def calc_DeltaE(self, rhs):
        sum = pow(self.L - rhs.L, 2)
        sum += pow(self.a - rhs.a, 2)
        sum += pow(self.b - rhs.b, 2);
        return pow(sum, 0.5);

    def __repr__(self):
        return "L=%1f a*=%1f b*=%1f" % (self.L, self.a, self.b)

if __name__ == "__main__":
    if len(sys.argv) != 7:
        sys.stderr.write("usage: Eab_calc <x-m> <y-m> <Y-m> <x-ref> <y-ref> <Y-ref>\n")
        sys.exit(1)
```

```
measured_data = Lab_set().init_with_xyY(float(sys.argv[1]),
                                         float(sys.argv[2]),
                                         float(sys.argv[3]))

reference_data = Lab_set().init_with_xyY(float(sys.argv[4]),
                                         float(sys.argv[5]),
                                         float(sys.argv[6]))

print " measured: %s" % (measured_data)
print "reference: %s" % (reference_data)
print "DeltaE=%.1f" % (reference_data.calc_DeltaE(measured_data))

#
# end Eab_calc.py
#
```

## C.7. uuid\_check.py

This program reads one or more XML files containing d-cinema metadata and tests each of the UUID values for compliance with [RFC 4122]. The program will halt with an exception if any unexpected values are encountered. The following example illustrates this usage for a KDM file:

### Example C.7. uuid\_check.py execution

```
$ uuid_check.py Example.kdm.xml
UUID: 7556bff9-58f9-4320-bb1f-fb594219a957
UUID: bdb3a717-5062-4822-8dfc-0dc6570cc116
UUID: 71f7926e-8ce6-4763-b14b-0ef7dcd952f5
UUID: 6083adad-472c-43da-b131-c6dc601cd154
UUID: aeaae312-a257-11da-a601-8b319b685f8e
```

### C.7.1. uuid\_check Source Code Listing

```
#!/usr/bin/python
#
# uuid_check.py -- Scan an XML file and see that all UUID values
#                  conform to RFC-4122
#
#
import sys, re

# regular expressions for use below
urn_uuid_re = re.compile('urn:uuid:([^\<]*)')
uuid_re = re.compile('^([0-9a-f]{8}-[0-9a-f]{4}-\
([1-5])[0-9a-f]{3}-[8-9a-b][0-9a-f]{3}-[0-9a-f]{12}$', re.IGNORECASE)

#
def uuid_scan(text):
    uuid_list = []
    while text:
        match = urn_uuid_re.search(text)
        if not match: break

        uuid_val = match.group(1)
        text = text[match.end():]

        match = uuid_re.match(uuid_val)
        if not match:
            sys.stderr.write("urn:uuid: value is not an RFC-4122 UUID: %s\n" % (uuid_val))
            continue

        type = int(match.group(1)[0])
        if type not in (1, 4, 5):
            sys.stderr.write("Unexpected UUID type: %d for value %s\n" % (type, uuid_val))

        uuid_list.append(uuid_val)

    return uuid_list

#
#
if len(sys.argv) < 2:
    sys.stderr.write("usage: uuid_check.py <xml-file> [...]\n")
    sys.exit(1)
```



Version  
NEWER  
Vers:

```

for filename in sys.argv[1:]:
    try:
        handle = open(filename)
        text = handle.read()
        handle.close()

    except Exception, e:
        print "%s: %s" % (filename, e)

    else:
        for uuid in uuid_scan(text):
            print "UUID: " + uuid

#
# end uuid_check.py
#

```

ONLY  
FILE  
9

## C.8. dsig\_cert.py

This program reads a signed XML file and re-writes the file to the standard output using the certificate order expected by the **checksig** from the XML Security package. The following example illustrates this usage for a KDM file:

### Example C.8. dsig\_cert.py execution

```
$ dsig-cert.py test-kdm.xml >tmp.xml
$ checksig tmp.xml
Signature verified OK!
```

### C.8.1. dsig\_cert.py Source Code Listing

```
#!/usr/bin/python
#
# dsig_cert.py -- Re-order certificates in an XML signature
#
# NOTE: This program requires Python 2.4 or greater
#
#

import sys, re
from subprocess import Popen, PIPE

# regular expressions for use below
SignatureValue_end_re = re.compile('</(?:[\w\-\-]+:)?SignatureValue[^\>]*>')
X509Data_re = re.compile('<(?:[\w\-\-]+:)?X509Data[^\>]*>(.*?)</(?:[\w\-\-]+:)?X509Data\s*>\s+',
                          re.DOTALL)
X509Certificate_re = re.compile('X509Certificate[^\>]*>(.*?)</', re.DOTALL)
dnQualifier_re = re.compile('dnQualifier=(([\w\-\-]+)')

#
def get_dnq_type(pem_text, type):
    """Extract the dnQualifier value for the given certificate and common name."""
    handle = Popen(['/usr/bin/openssl', 'x509', '-noout', '-'+type],
                  stdin=PIPE, stdout=PIPE, close_fds=True)

    handle.stdin.write(pem_text)
    handle.stdin.close()
    name_text = handle.stdout.read()
    handle.wait()

    if handle.returncode != 0:
        raise Exception, "No X509Certificate element in " + pem_text

    dnq = dnQualifier_re.search(name_text)
    if not dnq:
        raise Exception, "Error retrieving dnQualifier from %s." % type

    return dnq.group(1)

#
def PEMify(base64_text):
    """ create canonical PEM lines from any base64 input"""
    in_text = re.sub('[\r\n]', '', base64_text)
    idx = 0
    end = len(in_text)
    retval = ''
    while idx < end:
        retval += in_text[idx:idx+64] + '\n'
        idx += 64
```

```

return retval

#
class dsig_certificate_set:
    """An object for manipulating XML Signature certificates."""
    def __init__(self, xml_doc):
        """Initialize with a signed XML document string."""
        body_end = SignatureValue_end_re.search(xml_doc)

        if not body_end:
            raise Exception, "Document does not contain a SignatureValue element."

        self.kdm_head = xml_doc[:body_end.end()]
        xml_doc = xml_doc[body_end.end():]
        self.X509Data_list = []

        x509_data = X509Data_re.search(xml_doc)
        if x509_data:
            self.kdm_head += xml_doc[:x509_data.start()]

        while x509_data:
            x509_text = xml_doc[x509_data.start():x509_data.end()]
            self.X509Data_list.append({ 'text': x509_text })
            xml_doc = xml_doc[x509_data.end():]
            x509_data = X509Data_re.search(xml_doc)

        self.kdm_tail = xml_doc

        for x509_data in self.X509Data_list:
            # extract the certificate
            cert = X509Certificate_re.search(x509_data['text'])
            if not cert:
                raise Exception, "No X509Certificate element in " + x509_data['text']

            cert = PEMify(cert.group(1))
            cert = "-----BEGIN CERTIFICATE-----\n%s-----END CERTIFICATE-----\n" % (cert)

            x509_data['subject_dnq'] = get_dnq_type(cert, 'subject')
            x509_data['issuer_dnq'] = get_dnq_type(cert, 'issuer')
            x509_data['pem_cert'] = cert

    def order_by_dnq(self):
        """Arrange certificates in leaf-root order."""
        root = None
        issuer_map = {}

        for x509_data in self.X509Data_list:
            if x509_data['subject_dnq'] == x509_data['issuer_dnq']:
                if root:
                    raise Exception, "Certificate list contains multiple roots."
                root = x509_data
            else:
                issuer_map[x509_data['issuer_dnq']] = x509_data

        if not root:
            raise Exception, "Self-signed root certificate not found."

        tmp_list = [root];
        try:
            key = tmp_list[-1]['subject_dnq']
            next = issuer_map[key]
            while next:
                tmp_list.append(next)
                key = tmp_list[-1]['subject_dnq']
                next = issuer_map[key]
        except:

```

```
        pass

    if len(self.X509Data_list) != len(tmp_list):
        raise Exception, "Certificates do not form a complete chain."

    tmp_list.reverse()
    self.X509Data_list = tmp_list
    return self

def write_certs(self, prefix='cert_set_'):
    """Write PEMcertificates to files using the optional filename prefix value."""
    count = 1
    for x509_data in self.X509Data_list:
        filename = "%s%d.pem" % (prefix, count)
        handle = open(filename, 'w')
        handle.write(x509_data['pem_cert'])
        handle.close()
        count += 1

def __repr__(self):
    cert_text = ''
    for cert in self.X509Data_list:
        cert_text += cert['text']

    return self.kdm_head + cert_text + self.kdm_tail

#
if __name__ == '__main__':
    if len(sys.argv) < 2:
        sys.stderr.write("usage: dsig_cert.py <xml-file>\n")
        sys.exit(1)

    try:
        handle = open(sys.argv[1])
        text = handle.read()
        handle.close()

        set = dsig_certificate_set(text)
        set.order_by_dnq()
        print set
    except Exception, e:
        print e

#
# end dsig_cert.py
#
```

## C.9. dsig\_extract.py

This program reads a signed XML file and writes the certificates contained within to individual PEM files. As shown below, the `-p` option can be used to provide a prefix for the automatically-generated filenames. In this example, the input document contained four certificates.

### Example C.9. dsig\_extract.py execution

```
$ dsig-extract.py -p my_prefix_ test-kdm.xml
$ ls my_prefix_*
my_prefix_1.pem
my_prefix_2.pem
my_prefix_3.pem
my_prefix_4.pem
```

### C.9.1. dsig\_extract.py Source Code Listing

```
#!/usr/bin/python
#
# dsig_extract.py -- Extract certificates from an XML signature
#
#
from dsig_cert import dsig_certificate_set
import sys

prefix = 'xmldsig_cert_'
filename = None

def usage():
    sys.stderr.write("usage: dsig_extract.py [-p <prefix>] <xml-file>\n")
    sys.exit(1)

if len(sys.argv) < 2:
    usage()

if sys.argv[1] == '-p':
    if len(sys.argv) < 4:
        usage()
    prefix = sys.argv[2]
    filename = sys.argv[3]
else:
    filename = sys.argv[1]

try:
    handle = open(filename)
    text = handle.read()
    handle.close()

    set = dsig_certificate_set(text)
    set.write_certs(prefix=prefix)

except Exception, e:
    print e

#
# end dsig_extract.py
#
```

---

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

**Page Intentionally Left Blank**

# Appendix D. ASM Simulator

The **asm-requester** and **asm-responder** programs implement the Auditorium Security Message (ASM) protocol via TLS as defined in [SMPTE-430-6]. Both programs have options that are required for each invocation, e.g., to specify the TLS certificate, certificate chain, and RSA private key. In the examples in the text of this document, these are collectively referred to as (... standard options ...). Such command examples (illustrated below) are intended to illustrate the options that apply specifically to the use case at hand.

**asm-requester** performs ASM requester duties, *i.e.* sending request messages to an LDB. It has options to specify the destination IP address, plus the certificate, certificate chain, and RSA private key that comprise its identity. To configure its ASM functionality, it has options to send message requests according to the message types defined in [SMPTE-430-6], as well as options that control its message sending behavior. It also has options to specify a filename prefix to store received data in the local filesystem. The standard behavior of the program is to display received messages to the standard output device.

**asm-responder** performs ASM Responder duties, *i.e.*, responding to requests from an SM. It acts as an ASM responder at startup, logging events that occur from "boot time" (program invocation) through program termination. If the program receives a HUP signal it will display the contents of its LE key register and log message queue. The program understands message types defined in [SMPTE-430-6] which it accepts and responds to accordingly. To control TLS functionality, it has options to specify the IP bind address and TCP port, plus the certificate, certificate chain, and RSA private key that comprise its identity. To configure its ASM functionality, it has options to control its message response behavior, *e.g.*, causing the program to respond to all request messages with "Busy". Files containing XML messages can be specified at invocation to pre-load log events to be returned in response to GetEventList messages. Preloaded messages will have their date stamps and message IDs, and message sequence numbers replaced with appropriate values.

The **asm-requester** and **asm-responder** programs are not provided with this document. They are described in detail in this appendix to allow Testing Organizations and other interested parties to develop an implementation that can provide the services required to execute the respective test procedures defined in this document. In lieu of developing this program, interested parties may instead choose to instrument an existing ASM requester or ASM responder implementation.

## Name

asm-requester — initiate Request-Response-Pair (RRP) message type requests to an RRP responder

## Synopsis

```
asm-requester --responder-address <address> --certificate-file <cert.pem> --certificate-chain-file <chain.pem>  
--private-key-file <key.pem> [--captured-prefix <string>] [--pre-request-wait-time <seconds>]  
[--post-request-wait-time <seconds>] [--strict-response-times] [--messagetype-id <id-  
string>] [--messagetype <messagetype>] [--interval <seconds>] [--repeat-count <integer>]  
[--start-time <timestamp>] [--end-time <timestamp>]
```

## Description

**asm-requester** is an ASM requester simulator. For a responder at a specified IP address, it will issue ASM Request messages to a responder and receive the ASM Responses from the responder. **asm-requester** recognizes ASM message types specified by [SMPTE 430-6].

## Options

--responder-address <address> -- The address of the responder host.

- `--certificate-file <cert.pem>` -- The certificate used to initialize the TLS connection to the responder.
- `--certificate-chain-file <cert-chain.pem>` -- A file containing the concatenated certificates starting with the CA root and ending with the last signer certificate.
- `--private-key-file <private-key.pem>` -- A file containing the private key that corresponds to the certificate specified by `--certificate-file`.
- `--captured-prefix <string>` -- Specify a filename prefix for logging responses from a responder to a file. The default is to write received responses to standard output.
- `--pre-request-wait-time <seconds>` -- The time to wait, in seconds, after establishing the TLS session and before issuing requests to the responder.
- `--post-request-wait-time <seconds>` -- The time to wait, in seconds, after issuing requests to the responder but before terminating the TLS session.
- `--strict-response-times` -- Enforce 2 second response time as specified in SMPTE 430-6.
- `--messagetype-id <id-string>` -- Specify an ID to be used with messagetypes that set or request a response based on an ID.
- `--messagetype <messagetype>` -- Specify the messagetype of the RRP being initiated. Valid message types are listed in the MESSAGE TYPES section below.
- `--interval <seconds>` -- The interval, in seconds, between repetitions of commands. When absent, the messagetype request is sent only once.
- `--repeat-count <integer>` -- The number of times a message request is sent to the requester, subject to any specified wait periods or intervals.
- `--start-time <timestamp>` -- Specify the starting timestamp for retrieving events using `GetEventList`.
- `--end-time <timestamp>` -- Specify the end timestamp for retrieving events using `GetEventList`.

## Message Types

ASM messages types fit into two categories: General Purpose ASM commands and Link Encryption ASM commands. Only one command (messagetype request) can be specified at a time. The following list describes the ASM Responder message types are accepted by the `--messagetype` option:

<code>BadRequest</code>	Issues a request for an unknown message type to illicit a "BadRequest" response.
<code>GetTime</code>	Issues a request for the current time of the responder.
<code>GetEventList</code>	Issues a request for the list of events recorded between specified start and stop times.
<code>GetEventID</code>	Issues a request for the log record matching one of the log record IDs returned from a <code>GetEventList</code> response.
<code>QuerySPB</code>	Issues a request for a system status report from the responder.
<code>LEKeyLoad</code>	Issues an <code>LEKeyLoad</code> message containing a link decryption key to the responder.
<code>LEKeyQueryID</code>	Issues an <code>LEKeyQuery</code> message specifying the ID of a link decryption key



LEKeyQueryAll	Issues an LEKeyQueryALL message to a responder.
LEKeyPurgeID	Issues an LEKeyPurgeID message containing an ID of a key to be purged.
LEKeyPurgeAll	Issues an LEKeyPurgeAll message instructing a responder to purge all link decryption keys.

## Examples

```
$ asm-requester --responder-address 192.168.1.100 \  
  --certificate-file virt-sm-001-cert.pem \  
  --certificate-chain sm-co-cert-chain.pem \  
  --private-key virt-sm-001.key \  
  --captured-prefix virt-ldb-001-test01- \  
  --pre-request-wait-time 60 \  
  --messagetype GetEventList
```

Starts an instance of the `asm-requester` with the specified identity files, to establish a TLS connection to 192.168.1.100, wait 60 seconds, then send a `GetEventList` message request. Output is logged to a file starting with the filename "virt-ldb-001-test01-".

## Name

`asm-responder` — respond to Request-Response-Pair (RRP) messagetype requests from an RRP requester

## Synopsis

```
asm-responder --bind-address <address:port> --certificate-file <cert.pem> --certificate-chain-file <chain.pem>  
--private-key-file <key.pem> [--captured-prefix <string>] [--respond-with-busy] [--respond-with-invalid] [--preload-log-event <xml-file>] [--tls-only]
```

## Description

**asm-responder** is an ASM responder simulator. It will respond to ASM Request messages received from an ASM requester. **asm-responder** recognizes ASM message types specified by SMPTE 430-6.

## Options

`--bind-address <address:port>` -- The IP address and port number to bind to and listen for connections. If not address is specified, bind to all addresses. If a port is not specified, the default port of 1173 is used.

`--certificate-file <cert.pem>` -- The certificate used when initializing the TLS connection to the requester.

`--certificate-chain-file <cert-chain.pem>` -- A file containing the concatenated certificates starting with the CA root and ending with the last signer certificate.

`--private-key-file <private-key.pem>` -- A file containing the private key that corresponds to the certificate specified by `--certificate-file`.

`--captured-prefix <string>` -- Specify a filename prefix for logging responses from a responder to a file. The default is to write received responses to standard output.

`--respond-with <message-type>` -- Respond to all request messages with the specified messagetype, either "Busy" or "Invalid". This option overrides all other response message options.

`--respond-with-queryspb-type <message-type>` -- Respond to all "QuerySPB" request messages with the specified response, either "NotPlaying", "Playing", or "SecurityAlert".

`--preload-log-event <event-log-file.xml>` -- Specify a file containing a log event. This option may be used multiple times, but only a single file may be specified per use.

`--tls-only` -- This option causes **asm-responder** to establish a TLS session when requested, then ignore (not respond to) any messages sent from a requester.

## Message Types

ASM messages types fit into two categories: General Purpose ASM commands and Link Encryption ASM commands. Only one command (messagetype request) can be specified at a time. The following list describes the ASM Responder message types that are recognized by the responder, and the action and response generated by receiving each message type:

BadRequest	Respond with a "BadRequest" response.
GetTime	Responds with a GetTime response message.
GetEventList	Responds with a message containing the list of events recorded between the start and stop times.
GetEventID	Responds with a message containing the log record matching the log record ID specified in the Requester's request message.
QuerySPB	Responds with a message containing a system status report.
LEKeyLoad	Accepts a Link Encryption key and responds with a message indicating that the key was received.
LEKeyQueryID	Responds with a message indicating the presence or absence of the key matching the KeyID specified in the requester's message.
LEKeyQueryAll	Responds with a message containing all of the KeyIDs corresponding to the Link Decryption keys in the responder.
LEKeyPurgeID	Deletes the specified key, and responds with a message indicating that the key matching the KeyID specified by the requester has been deleted.
LEKeyPurgeAll	Deletes all Link Encryption keys and responds with a message indicating that the key matching the KeyID specified by the requester has been deleted.

## Examples

```
$ asm-responder --bind-address 192.168.1.100:1173
--certificate-file virt-ldb-001-cert.pem \
--certificate-chain ldb-co-cert-chain.pem \
--private-key virt-ldb-001.key
```

Invokes **asm-responder** with an identity, and a listening address and port.

```
$ asm-responder --bind-address 192.168.1.100:1173
--certificate-file virt-ldb-001-cert.pem \
--certificate-chain ldb-co-cert-chain.pem \
--private-key virt-ldb-001.key \
--respond-with Busy
```

---

Invokes **asm-responder** with an identity, and a listening address and port. This invocation is configured to respond to all message requests with a "ResponderBusy" response.

**Page Intentionally Left Blank**

# Appendix E. GPIO Test Fixture

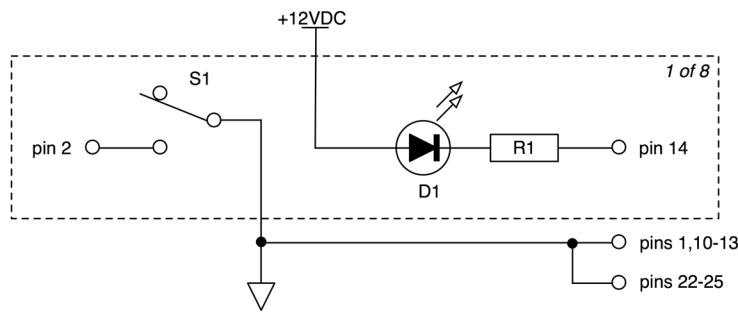
The GPIO test fixture has eight outputs, which connect to ground via normally-open switch contacts. These outputs are expected to interface to command and/or status inputs of the d-cinema equipment under test.

The fixture has eight inputs, which connect to powered, current limited LEDs and will illuminate when the corresponding input is grounded. These inputs interface to command and/or status outputs of the d-cinema equipment under test.

Example circuits are provided below. Interface of outputs, inputs and ground is made via a single DB-25 female connector on the test fixture.

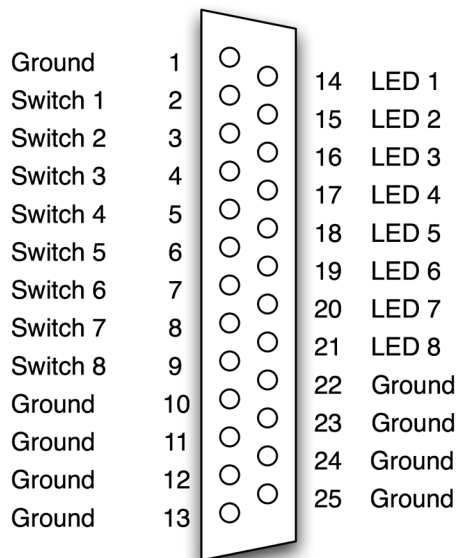
Testing Entities are not required to follow the above design, and are free to develop their own equipment and connector standards. The manufacturer of the d-cinema equipment being tested is responsible for providing a cable, appropriate for the individual device under test, that will interface to the test fixture being used.

**Figure E.1. GPIO Test Fixture Schematic**



Note that the LED inputs are internally current limited. External devices will be expected to sink 25mA per channel. Also, the test fixture has an integral PSU (the PSU may be external but it must use a different connector).

**Figure E.2. GPIO Test Fixture Connector**



**Page Intentionally Left Blank**

---

# Appendix F. Reference Documents

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 2009

- [AES3-2003] AES standard for digital audio - Digital input-output interfacing - Serial transmission format for two-channel linearly represented digital audio data, Audio Engineering Society, September 9, 2003
- [CIE-15-2004] Colorimetry, 3rd Edition, International Commission on Illumination, 2004
- [DCI-DCSS-1-1] Digital Cinema System Specification. Version 1.1, DCI, April 12, 2007
- [FIPS-140-2] FIPS 140-2: Security Requirements for Cryptographic Modules, NIST, May 25, 2001
- [FIPS-180-2] FIPS 180-2: Secure Hash Standard, NIST, August 1, 2002
- [FIPS-197] FIPS 197: Advanced Encryption Standard (AES), NIST, November 26, 2001
- [FIPS-198a] FIPS 198a: The Keyed-Hash Message Authentication Code (HMAC), NIST, March 6, 2002
- [IEEE-802-3] 802-3: Standard for Information technology—Telecommunications and information exchange between systems— Local and metropolitan area networks—Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, IEEE, 2005
- [ISO-144496] ISO/IEC 144496: Information technology - Computer graphics and image processing - Font Compression and Streaming, ISO/IEC, 2004
- [ISO-15948] ISO-15948: Information technology - Computer graphics and image processing - Portable Network Graphics (PNG): Functional specification, ISO/IEC, 2004
- [ISO-15444-1] ISO/IEC 15444-1 2004, Information Technology: JPEG 2000 Image Coding System, ISO/IEC, 2004
- [ISO-15444-1-AMD-1] ISO/IEC 15444-1/Amd1:2006 Codestream restrictions - Amendment 1: Profiles for digital cinema applications, ISO/IEC, 2004
- [ISO-10646] ISO 10646: Information technology – Universal Multiple-Octet Coded Character Set (UCS), ISO/IEC, December 15, 2003
- [ITU-X509] ITU-T Recommendation X.509 (1997 E): Information Technology – Open Systems Interconnection – The Directory: Authentication Framework, ITU, June 1997
- [NIST-800-38A] NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation - Methods and Techniques, NIST, Morris Dworkin, December 2001
- [PKCS-1] PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002
- [RFC-1421] RFC 1421: "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", IETF, J. Linn, February 1993
- [RFC-2246] RFC 2246: "The TLS Protocol Version 1.0", IETF, T. Dierks, C. Allen, January 1999
- [RFC-2253] RFC 2253: "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", IETF,
- [RFC-3174] RFC 3174: "US Secure Hash Algorithm 1 (SHA1)", IETF, September 2001

- [RFC-3339] RFC 3339: "Date and Time on the Internet: Timestamps", IETF, July 2002
- [RFC-3447] RFC 3447: "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", IETF, J. Jonsson, B. Kaliski, February 2003
- [RFC-4086] RFC 4086: "Randomness Requirements for Security", IETF, D. Eastlake 3rd, J. Schiller, S. Crocker, June 2005
- [RFC-4122] RFC 4122: "A Universally Unique Identifier (UUID) URN Namespace", IETF, P. Leach, M. Mealling, July 2005
- [SMPTE-12M] SMPTE 12M: "Standard for Television, Audio and Film-Time and Control Code", SMPTE, 1999
- [SMPTE-EG41] SMPTE EG41 "Material Exchange Format (MXF) – Engineering Guideline", SMPTE, 2004
- [SMPTE-EG42] SMPTE EG42 "Material Exchange Format (MXF) – MXF Descriptive Metadata", SMPTE, 2004
- [SMPTE-330M] SMPTE 330M-2004: "SMPTE Standard for Television – Unique Material Identifier (UMID)", SMPTE, February 3, 2004
- [SMPTE-336M] SMPTE 336M-2001: "SMPTE Standard for Television - Data Encoding Protocol using Key-Length-Value", SMPTE, 2001
- [SMPTE-372M] SMPTE 372M: "Dual Link 292M Interface for 1920 x 1080 Picture Raster", SMPTE, 2002
- [SMPTE-377M-2004] SMPTE 377M-2004: "SMPTE Standard for Television – Material Exchange Format (MXF) – File Format Specification (Standard)", SMPTE, September 22, 2004
- [SMPTE-379M-2004] SMPTE 379M-2004: "SMPTE Standard for Television – Material Exchange Format (MXF) – MXF Generic Container", SMPTE, September 22, 2004
- [SMPTE-382M] (draft) SMPTE 382M, "Material Exchange Format (MXF) – Mapping AES3 and Broadcast Wave Audio into the MXF Generic Container", SMPTE, n/a
- [SMPTE-422M-2006] SMPTE 422M-2006, "Material Exchange Format - Mapping JPEG 2000 Codestreams into the MXF Generic Container", SMPTE, August 28, 2006
- [SMPTE-410M] draft SMPTE 410M, "Material Exchange Format - Generic Stream Partition", SMPTE, August, 2007
- [SMPTE-428-1] SMPTE 428-1, "D-Cinema Distribution Master Image Structure", SMPTE, February 01, 2006
- [SMPTE-428-2] SMPTE 428-2, "D-Cinema Distribution Master- Audio Characteristics", SMPTE, October 27, 2005
- [SMPTE-428-3] SMPTE 428-3, "D-Cinema Distribution Master Audio Channel Mapping", SMPTE, 2006
- [SMPTE-428-7] SMPTE 428-7 2007, "D-Cinema Distribution Master - Subtitle", SMPTE, 2007
- [SMPTE-429-2] Draft SMPTE 429-2, "D-Cinema Packaging Operational Constraints", SMPTE, September 2007
- [SMPTE-429-3] SMPTE 429-3, "D-Cinema Packaging Sound and Picture Track File Application", SMPTE, February 27, 2006



- 
- Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8, 2009
- [SMPTE-429-4] SMPTE 429-4, "D-Cinema Packaging MXF JPEG2000 Application", SMPTE, February 09, 2006
- [SMPTE-429-5] draft SMPTE 429-5, "D-Cinema Packaging Subtitling Distribution Format", SMPTE, June, 2007
- [SMPTE-429-6] SMPTE 429-6, "D-Cinema Packaging Track File Essence Encryption", SMPTE, March 13, 2006
- [SMPTE-429-7] SMPTE 429-7, "D-Cinema Packaging Composition Playlist Application", SMPTE, March 03, 2006
- [SMPTE-429-8-2007] SMPTE 429-8-2007, "D-Cinema Packaging Packing List", SMPTE, July 6, 2007
- [SMPTE-429-9-2007] SMPTE 429-9-2007, "D-Cinema Packaging Asset Mapping", SMPTE, July 6, 2007
- [SMPTE-430-1] SMPTE 430-1, "D-Cinema Operations Key Delivery Message", SMPTE, March 13, 2005
- [SMPTE-430-2] SMPTE 430-2, "D-Cinema Operations Digital Certificates", SMPTE, March 13, 2005
- [SMPTE-430-3] SMPTE 430-3, "D-Cinema Operations Extra-Theatre Messaging", SMPTE, March 3, 2005
- [SMPTE-430-4] draft SMPTE 430-4, "D-Cinema Operations Log Records Format", SMPTE, July 2, 2006
- [SMPTE-430-5] draft SMPTE 430-5, "D-Cinema Operations Log Records Security Constraints", SMPTE, July 2, 2006
- [SMPTE-430-6] SMPTE 430-6-2007, "Auditorium Security Messages for Intra-Theater Communications", SMPTE, August 9, 2006
- [SMPTE-431-1] SMPTE 431-1, "D-Cinema Quality Screen Luminance Level, Chromaticity, and Uniformity", SMPTE,
- [SMPTE-431-2] SMPTE RP 431-2-2007, "D-Cinema Quality - Reference Projector and Environment", SMPTE, July 27, 2007
- [SMPTE-431-3] SMPTE 431-3, "D-Cinema Quality Projection Image Measurements", SMPTE, May 31, 2005
- [SMPTE-432-1] SMPTE EG 432-1-2007, "Digital Source Processing - Color Processing for D-Cinema", SMPTE, April 12, 2007
- [SMPTE-432-2] SMPTE 432-2, "D-Cinema Signal Processing Audio LFE Characteristics", SMPTE, September 13, 2005
- [SMPTE-433M] SMPTE 433-1, "D-Cinema XML Data Types", SMPTE, 2007

**Page Intentionally Left Blank**

# Appendix G. DCI Specification v1.1

## References to CTP

Version 1.0 ARCHIVE ONLY  
 NEWER VERSION AVAILABLE  
 Version 1.1 released May 8 2009

DCSS V1.1 Section	Procedure Title	CTP Section	CTP Page
3.2.1.2	Image Structure Container and Image Container Format	4.5.1	118
3.2.1.3	Image Structure Container and Image Container Format	4.5.1	118
3.2.1.5	Image Compression Standard & Encoding Parameters	4.5.2	120
3.2.1.7	Image Structure Container and Image Container Format	4.5.1	118
3.3.2.2	Audio Characteristics	4.5.3	122
3.3.2.3	Audio Sample Rate Conversion	6.6.2	206
3.3.4	Timed Text Track File Format	4.4.3	111
3.3.4.1	Audio Characteristics	4.5.3	122
3.4.2.2	Timed Text Resource Encoding	4.5.4	124
3.4.2.5	Timed Text Synchronization	6.7.2	211
3.4.3.4	Timed Text Resource Encoding	4.5.4	124
3.4.3.7	Support for Multiple Captions	6.7.3	212
3.4.3.8	Timed Text Synchronization	6.7.2	211
4.2	Image Compression Standard & Encoding Parameters	4.5.2	120
4.3.2	Decoder Requirements	6.5.2	202
4.4	Image Compression Standard & Encoding Parameters	4.5.2	120
4.4.3.2	Timed Text Resource Encoding	4.5.4	124
5.2.2.2	Image and Audio Packaging Standard	4.4.2	109
5.2.2.3	Image and Audio Packaging Standard	4.4.2	109
5.2.2.4	Image and Audio Packaging Standard	4.4.2	109
5.2.2.5	Image and Audio Packaging Standard	4.4.2	109
5.2.2.6	Image and Audio Packaging Standard	4.4.2	109
5.2.3	Composition Playlist File	4.3.1	100
	Composition Playlist Signature Validation	4.3.2	102
5.3.1.10	Image and Audio Packaging Standard	4.4.2	109
5.3.1.11	Image and Audio Packaging Standard	4.4.2	109
5.3.1.12	Image and Audio Packaging Standard	4.4.2	109
5.3.1.2	Image and Audio Packaging Standard	4.4.2	109
5.3.1.3	Track File Length	4.4.4	113
	Playback of Image Only Material	6.5.1	201
5.3.1.6	Click Free Splicing of Audio Track Files	6.6.4	209
5.3.1.7	Track File Key Epoch	4.3.3	103

<b>DCSS V1.1 Section</b>	<b>Procedure Title</b>	<b>CTP Section</b>	<b>CTP Page</b>
5.3.3.2	Image Track File Frame Boundary	4.4.5	114
5.3.4.2	Audio Track File Frame Boundary	4.4.6	116
5.4.2	Composition Playlist File	4.3.1	100
5.4.3	Composition Playlist File	4.3.1	100
5.4.3.2	Composition Playlist File	4.3.1	100
5.4.3.3	Composition Playlist File	4.3.1	100
5.4.3.4	Composition Playlist File	4.3.1	100
5.4.3.6	Composition Playlist Signature Validation	4.3.2	102
5.4.4	Composition Playlist Signature Validation	4.3.2	102
5.5.2.1	Asset Map File	4.1.1	93
	Volume Index File	4.1.2	94
5.5.2.3	Packing List Signature Validation	4.2.2	98
5.5.3.1	Packing List File	4.2.1	96
5.5.3.2	Packing List File	4.2.1	96
	Packing List Signature Validation	4.2.2	98
6.2.3	Storage System Ingest Interface	8.1.1	249
7.2.2	Screen Management System	8.2	253
7.2.3.1	Theater System Reliability	10.4.1	281
7.2.3.11	Storage System Capacity	8.1.2	250
7.2.3.13	Restarting Playback	8.2.8	261
7.2.3.2	Theater System Reliability	10.4.1	281
7.2.3.5	Show Playlist Creation	8.2.2	254
7.2.3.7	Show Playlist Creation	8.2.2	254
7.3.2	Show Playlist Format	8.2.3	256
7.3.3.1	Show Playlist Format	8.2.3	256
7.3.3.2	Show Playlist Format	8.2.3	256
7.3.4	Show Playlist Creation	8.2.2	254
	Automation Control and Interfaces	8.2.5	258
7.4.1.1	Show Playlist Creation	8.2.2	254
7.4.1.2	Show Playlist Creation	8.2.2	254
	Restarting Playback	8.2.8	261
7.4.1.3	Show Playlist Creation	8.2.2	254
	SMS User Accounts	8.2.9	262
7.4.1.4	Show Playlist Creation	8.2.2	254
7.4.1.5	Show Playlist Creation	8.2.2	254
7.4.1.6	Show Playlist Creation	8.2.2	254
	KDM Validity Checks	8.2.4	257

DCSS V1.1 Section	Procedure Title	CTP Section	CTP Page
	Automation Control and Interfaces	8.2.5	258
	Artifact Free Playback of Image Content	8.2.7	260
7.4.1.7	Automation Control and Interfaces	8.2.5	258
7.4.1.8	Interrupt Free Playback	8.2.6	259
	Restarting Playback	8.2.8	261
	Audio Delay Setup	6.6.3	207
7.5.3.2	Storage System Redundancy	8.1.3	251
7.5.3.3	Storage System Performance	8.1.4	252
7.5.3.4	Storage System Performance	8.1.4	252
7.5.3.6	Storage System Performance	8.1.4	252
7.5.3.8	Theater System Storage Security	10.4.2	281
7.5.4.2.5	Media Block Overlay	6.7.1	210
7.5.4.2.6	Media Block Overlay	6.7.1	210
7.5.4.2.7	Media Block Overlay	6.7.1	210
7.5.6.1	Digital Audio Interfaces	6.6.1	204
7.5.6.2	Digital Audio Interfaces	6.6.1	204
7.5.7.2	Automation Control and Interfaces	8.2.5	258
7.5.8	Screen Management System	8.2	253
8.2.2.10	IMB Link Encryption	6.2.4	194
8.2.2.4	Projector Lens	7.5.2	237
8.2.2.6	Projector Pixel Count/Structure	7.5.3	238
8.2.2.7	Projector Pixel Count/Structure	7.5.3	238
	Projector Spatial Resolution and Frame Rate Conversion	7.5.4	239
8.2.2.8	Projector Spatial Resolution and Frame Rate Conversion	7.5.4	239
8.3.4.10	Contouring	7.5.10	245
8.3.4.11	Transfer Function	7.5.11	246
8.3.4.3	White Point Luminance and Uniformity	7.5.5	240
8.3.4.4	White Point Luminance and Uniformity	7.5.5	240
8.3.4.5	White Point Chromaticity and Uniformity	7.5.6	241
8.3.4.6	White Point Chromaticity and Uniformity	7.5.6	241
8.3.4.7	Sequential Contrast	7.5.7	242
8.3.4.8	Intra-frame Contrast	7.5.8	243
8.3.4.9	Grayscale Tracking	7.5.9	244
8.4.2	IMB Link Encryption	6.2.4	194
8.4.3.1	IMB Link Encryption	6.2.4	194
8.4.4	Projector Overlay	7.5.1	236
9.4.1	Security Devices Self-Test Capabilities	10.4.3	281

DCSS V1.1 Section	Procedure Title	CTP Section	CTP Page
9.4.1.1	SMS Operator Identification	8.2.10	263
	Security Entity Physical Protection	10.4.4	281
	Secure SMS-SM Communication	10.4.5	282
9.4.2.2	Projector Physical Protection	7.2.1	216
9.4.2.4	SM Operating Environment	9.5.1	274
	Location of Security Manager	10.4.6	282
	SM Usage of OS Security Features	10.4.7	282
	SM Secure Communications	10.4.8	282
9.4.2.5	SMS Identity and Certificate	8.2.11	264
9.4.3.5	Image Integrity Checking	6.1.1	181
	Sound Integrity Checking	6.1.2	183
	Restriction of Keying to Monitored Link Decryptors	6.1.3	184
	Restriction of Keying to MD Type	6.1.4	185
	Restriction of Keying to valid CPLs	6.1.5	186
	Remote SPB Integrity Monitoring	6.1.6	187
	SPB Integrity Fault Consequences	6.1.7	189
	LE Key Generation	9.5.2	274
	Location of Security Manager	10.4.6	282
	Playback Preparation	10.4.9	283
	SE Uniqueness Constraint	10.4.10	283
	Prevention of Keying of Compromised SPBs	10.4.11	283
	SPB Authentication	10.4.12	283
	TLS Session Key Refreshes	10.4.13	283
	LE Key Issuance	10.4.14	284
	Maximum Key Validity Period	10.4.15	284
KDM Purge upon Expiry	10.4.16	284	
Key Usage Time Window	10.4.17	284	
9.4.3.6.1	Projector Physical Protection	7.2.1	216
	Projector Access Door	7.2.2	217
	Electronic Marriage Break Key Retaining	7.2.8	223
	Projector Companion SPB Location	7.3.1	224
	Companion SPBs with Electronic Marriage	7.3.2	225
	Projector Secure Silicon Device	10.4.18	285
	Access to Projector Image Signals	10.4.19	285
	Systems with Electronic Marriage	10.4.20	285
9.4.3.6.2	Companion SPBs with Electronic Marriage	7.3.2	225
	Companion SPB Marriage Break Key Retaining	7.3.3	227

DCSS V1.1 Section	Procedure Title	CTP Section	CTP Page
	LDB TLS Session Constraints	7.4.2	229
	LDB Time-Awareness	7.4.3	230
	LDB ASM Conformity	7.4.4	231
	LDB Key Storage	7.4.5	232
	LDB Key Purging	7.4.6	233
	LDB Logging	7.4.7	235
	SPB1 Tamper Responsiveness	9.5.3	274
9.4.3.6.2.1	SPB1 Tamper Responsiveness	9.5.3	274
9.4.3.6.3	Companion SPBs with Electronic Marriage	7.3.2	225
	Companion SPB Marriage Break Key Retaining	7.3.3	227
	SPB1 Tamper Responsiveness	9.5.3	274
9.4.3.6.6	Systems Without Electronic Marriage	7.2.7	222
	LDB without Electronic Marriage	7.4.1	228
	Systems Without Electronic Marriage	10.4.21	285
9.4.3.7	Clock Adjustment	6.3.1	195
	Clock Battery	6.3.2	196
	Clock Resolution	6.3.3	197
	Clock Date-Time-Range	10.4.22	285
	Clock Setup	10.4.23	286
	Clock Stability	10.4.24	286
9.4.4	LDB Trust	6.2.1	190
	LE Key Usage	6.2.3	193
	LE Key Generation	9.5.2	274
9.4.4.1	Multiple LE Operation	6.2.2	191
	SE Uniqueness Constraint	10.4.10	283
9.4.4.7	Clock Continuity	10.4.27	286
9.4.5	ASM "RRP Invalid"	5.2.2.3	138
	ASM "GetTime"	5.2.2.4	139
	ASM "GetEventList"	5.2.2.5	140
	ASM "GetEventID"	5.2.2.6	141
	ASM "LEKeyLoad"	5.2.2.7	143
	ASM "LEKeyQueryID"	5.2.2.8	145
	ASM "LEKeyQueryAll"	5.2.2.9	146
	ASM "LEKeyPurgeID"	5.2.2.10	147
	ASM "LEKeyPurgeAll"	5.2.2.11	148
9.4.5.1	SM Secure Communications	10.4.8	282
	TLS Endpoints	10.4.28	287

DCSS V1.1 Section	Procedure Title	CTP Section	CTP Page
	Implementation of RRP	10.4.29	287
	SMS and SPB Authentication and ITM Transport Layer	10.4.30	287
9.4.5.2.1	Idempotency of ITM RRP	10.4.31	287
9.4.5.2.3	TLS Session Initiation	5.2.1	132
	Auditorium Security Message Support	5.2.2.1	134
	Security Design Description Requirements	9.5.4	275
	RRP Synchronism	10.4.32	287
	TLS Mode Bypass Prohibition	10.4.33	288
	RRP Broadcast Prohibition	10.4.34	288
	Implementation of Proprietary ITMs	10.4.35	288
	RRP Initiator	10.4.36	288
	SPB TLS Session Partners	10.4.37	288
	SM TLS Session Partners	10.4.38	289
	RRP "Busy" and unsupported Types	10.4.39	289
9.4.5.3.2	RRP Operational Message Ports	10.4.40	289
	LDB ASM Conformity	7.4.4	231
9.4.6.1.1	ASM Failure Behavior	5.2.2.2	136
	FM Payload	6.4.3	200
	FM Generic Inserter Requirements	10.4.41	289
	FM Algorithm General Requirements	10.4.42	290
9.4.6.1.2	FM Insertion Requirements	10.4.43	290
	IFM Visual Transparency	10.4.44	290
	IFM Robustness	10.4.45	290
9.4.6.1.3	AFM Inaudibility	10.4.46	291
	AFM Robustness	10.4.47	291
9.4.6.2	FM Application Constraints	6.4.1	198
	Granularity of FM Control	6.4.2	199
	FM Control Instance	10.4.48	291
9.4.6.3.1	Log Records for Multiple SPBs	5.3.2.2	154
	Log Collection by the SM	5.3.2.4	156
	SE Time Stamping	10.4.49	291
	SE Log Authoring	10.4.50	291
	SPB Log Storage Requirements	10.4.51	292
	Remote SPB Log Storage Requirements	10.4.52	292
	IMB Log Storage Capabilities	10.4.53	292
Logging for Standalone Systems	10.4.54	292	
9.4.6.3.10	General Log System Failure	5.3.2.5	157



DCSS V1.1 Section	Procedure Title	CTP Section	CTP Page
	SPB Log Failure	10.4.56	293
	Log Purging in failed SPBs	10.4.57	293
9.4.6.3.2	Log Structure	5.3.2.1	153
9.4.6.3.4	Log Sequence Numbers	5.3.2.3	155
9.4.6.3.7	SM Proxy of Log Events	5.3.3.1	158
	SM Proxy of Security Operations Events	5.3.3.2	159
	SM Proxy of Security ASM Events	5.3.3.3	161
	FrameSequencePlayed Event	5.4.1.1	163
	CPLStart Event	5.4.1.2	164
	CPLEnd Event	5.4.1.3	165
	PlayoutComplete Event	5.4.1.4	166
	CPLCheck Event	5.4.1.5	167
	KDMKeysReceived Event	5.4.1.6	168
	KDMDeleted Event	5.4.1.7	169
	LinkOpened Event	5.4.2.1	170
	LinkClosed Event	5.4.2.2	171
	LinkException Event	5.4.2.3	172
	LogTransfer Event	5.4.2.4	173
	KeyTransfer Event	5.4.2.5	174
	SPBStartup SPBShutdown Events	5.4.2.6	175
	SPBOpen SPBClose Events	5.4.2.7	176
	SPBClockadjust Event	5.4.2.8	177
	SPBMarriage and SPBDivorce Events	5.4.2.9	178
	Logging of Failed Procedures	10.4.55	292
9.4.6.6.3	IMB Tasks	10.4.58	293
9.5.1	SPB Digital Certificate	5.1.1	129
9.5.2.1	SPB2 Secure Silicon Requirements	7.2.4	219
	SPB2 Tamper Evidence	7.2.5	220
	SPB1 Tamper Responsiveness	9.5.3	274
	SPB1 Tamper Resistance	9.5.5	275
	SPB1 FIPS Requirements	9.5.6	275
9.5.2.2	SPB2 Secure Silicon Requirements	7.2.4	219
	SPB Type 2 Secure Silicon	5.1.3	131
	SPB1 Tamper Responsiveness	9.5.3	274
	SPB1 Tamper Resistance	9.5.5	275
	SPB1 Secure Silicon FIPS Requirements	9.5.7	275
	Private Keys outside Secure Silicon	10.4.59	293

DCSS V1.1 Section	Procedure Title	CTP Section	CTP Page
	Image Keys Outside Secure Silicon	10.4.60	293
9.5.2.3	SPB2 Secure Silicon Field Replacement	7.2.6	221
	Repair and Renewal of SPBs	10.4.25	286
	Prohibition of SPB1 Field Serviceability	10.4.61	294
	Use of Software Protection Methods	10.4.62	294
9.5.2.4	Projector Physical Protection	7.2.1	216
	SPB2 Requirements	7.2.3	218
	SPB Type 2 Security Perimeter	5.1.2	130
	SPB2 Protected Devices	10.4.26	286
9.5.2.5	SPB2 Secure Silicon Requirements	7.2.4	219
	SM Operating Environment	9.5.1	274
	LE Key Generation	9.5.2	274
	SPB1 Tamper Responsiveness	9.5.3	274
	Security Design Description Requirements	9.5.4	275
	SPB1 Tamper Resistance	9.5.5	275
	SPB1 FIPS Requirements	9.5.6	275
	SPB1 Secure Silicon FIPS Requirements	9.5.7	275
	Asymmetric Key Generation	9.5.8	276
	Critical Security Parameter Protection	9.5.9	276
	TMS role	10.4.63	294
9.5.2.6	SPB1 Tamper Responsiveness	9.5.3	274
	Critical Security Parameter Protection	9.5.9	276
	D-Cinema Security Parameter Protection	10.4.64	294
9.5.2.7	SM Operating Environment	9.5.1	274
	SPB 1 Firmware Modifications	9.5.10	276
9.6.1	Location of Security Manager	10.4.6	282
9.6.1.2	Location of Security Manager	10.4.6	282
9.7.5	Composition Playlist Signature Validation	4.3.2	102
9.7.6	RSA Key Format	2.1.6	16
	LE Key Generation	9.5.2	274
	Asymmetric Key Generation	9.5.8	276
	RSA Key Entropy	10.4.65	294
	Preloaded Symmetric Key Entropy	10.4.66	295
9.7.7	Maximum Number of DCP Keys	3.5.3	85
	Composition Playlist File	4.3.1	100
	MD Caching of Keys	10.4.67	295
9.8	Basic Certificate Structure	2.1.1	11

DCSS V1.1 Section	Procedure Title	CTP Section	CTP Page
	SignatureAlgorithm Fields	2.1.2	12
	SignatureValue Field	2.1.3	13
	SerialNumber Field	2.1.4	14
	SubjectPublicKeyInfo Field	2.1.5	15
	Validity Field	2.1.7	17
	AuthorityKeyIdentifier Field	2.1.8	18
	KeyUsage Field	2.1.9	19
	Basic Constraint Field	2.1.10	20
	Public Key Thumbprint	2.1.11	21
	Organization Name Field	2.1.12	23
	OrganizationUnitName Field	2.1.13	24
	Entity Name and Roles Field	2.1.14	25
	Unrecognized Extensions	2.1.15	26
	Signature Validation	2.1.16	27
	Certificate Chains	2.1.17	28
	ASN.1 DER Encoding Check	2.2.1	30
	Missing Required Fields	2.2.2	31
	PathLen Check	2.2.3	33
	OrganizationName Match Check	2.2.4	35
	Certificate Role Check	2.2.5	36
	Validity Date Check	2.2.6	37
	Signature Algorithm Check	2.2.7	38
	Public Key Type Check	2.2.8	39
	ETM Structure	3.3.1	50
	ETM Validity Date Check	3.3.2	51
	ETM Signer Element	3.3.3	52
	ETM EncryptionMethod Element	3.3.4	53
	KDM MessageType Element	3.4.1	62
	KDM SubjectName Element	3.4.2	63
	KDM ContentAuthenticator Element	3.4.3	64
	KDM KeyIdList/TypedKeyId Field	3.4.5	66
	KDM EncryptedData Element	3.4.7	68
	KeyInfo Element	3.4.8	69
	KDM DeviceListDescription Element	3.4.9	70
	KDM CompositionPlaylistId Element	3.4.13	74
	KDM Validity Fields	3.4.14	75
	KDM KeyIdList Element	3.4.15	76

Version 1.1 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

<b>DCSS V1.1 Section</b>	<b>Procedure Title</b>	<b>CTP Section</b>	<b>CTP Page</b>
	KDM CipherData Structure ID	3.4.16	77
	KDM CipherData Signer Thumbprint	3.4.17	78
	KDM CipherData Validity	3.4.18	79
	KDM CipherData CPL ID	3.4.19	80
	KDM NonCriticalExtensions Element	3.5.1	83
	ETM IssueDate Field Check	3.5.2	84
	Certificate Thumbprint Check	3.5.5	87

---

# Appendix H. Abbreviations

AES	Advanced Encryption Standard
CPL	Composition PlayList
DCI	Digital Cinema Initiatives, LLC
DCDM	Digital Cinema Distribution Master
DCP	Digital Cinema Package
DSM	Digital Source Master
ETM	Extra Theater Message
FM	Forensic Marking
GUI	Graphical User Interface
IMB	Image Media Block
ISO	International Organization for Standards
IETF	Internet Engineering Task Force
J2K	JPEG2000
KDM	Key Delivery Message
LCD	Liquid Crystal Display
LD	Link Decryptor
LDB	Link Decryptor Block
LE	Link Encryptor
MB	Media Block
MD	Media Decryptor
POSIX	Portable Operating System Interface
RAID	Redundant Array of Independent Disks (formerly: Redundant Array of Inexpensive Disks)
SM	Security Manager
SMS	Screen Management System
SPB	Secure Processing Block
SPL	Show Play List
SMPTE	Society of Motion Picture and Television Engineers

---

StEM	Standard Evaluation Material
TCP/IP	Transmission Control Protocol/Internet Protocol
TDL	Trusted Device List
TLS	Transport Layer Security (formerly known as SSL (Secure Socket Layer)
TMS	Theater Management Station
UMID	Unique Material Identifier
USB	Universal Serial Bus
UUID	Universally Unique Identifier
XML	eXtensible Markup Language

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

---

# Index

## A

Abbreviations, 451  
Access to Projector Image Signals, 285  
AFM Inaudibility, 291  
AFM Robustness, 291  
Artifact Free Playback of Image Content, 260  
ASM "GetEventID", 141  
ASM "GetEventList", 140  
ASM "GetTime", 139  
ASM "LEKeyLoad", 143  
ASM "LEKeyPurgeAll", 148  
ASM "LEKeyPurgeID", 147  
ASM "LEKeyQueryAll", 146  
ASM "LEKeyQueryID", 145  
ASM "RRP Invalid", 138  
ASM Failure Behavior, 136  
ASM Simulator  
    Requester, 429  
ASN.1 DER Encoding Check, 30  
Asset Map File, 93  
Asymmetric Key Generation, 276  
Audio Characteristics, 122  
Audio Delay Setup, 207  
Audio Reproduction, 204  
Audio Sample Rate Conversion, 206  
Audio Track File Frame Boundary, 116  
Auditorium Security Message, 134  
Auditorium Security Message Support, 134  
AuthorityKeyIdentifier Field, 18  
Automation Control and Interfaces, 258

## B

Basic Certificate Structure, 11  
Basic Constraint Field, 20

## C

Certificate Chains, 28  
Certificate Presence Check, 88  
Certificate Role Check, 36  
Certificate Thumbprint Check, 87  
Certificates  
    Certificate Decoder Behavior, 30  
    Testing, 9  
Click Free Splicing of Audio Track Files, 209  
Clock Adjustment, 195  
Clock Battery, 196  
Clock Continuity, 286  
Clock Date-Time-Range, 285  
Clock Resolution, 197

Clock Setup, 286  
Clock Stability, 286  
Clocks and Time, 195  
Color Accuracy, 247  
Companion SPB Marriage Break Key Retaining, 227  
Companion SPB Type 1, 224  
Companion SPBs with Electronic Marriage, 225  
Composition Playlist File, 100  
Composition Playlist Signature Validation, 102  
Content Keys and TDL check, 265  
Contouring, 245  
Control Messages  
    ETM, 50  
    KDM, 62  
    KDM Decoder Behavior, 83  
CPLCheck Event, 167  
CPEnd Event, 165  
CPLStart Event, 164  
Critical Security Parameter Protection, 276

## D

D-Cinema Security Parameter Protection, 294  
DCI Requirements Review, 279  
DCI Specification v1.1 References to CTP, 441  
DCP, 91  
    Asset Map, 91  
    Composition Playlist, 100  
    Consolidated Test Sequence, 301  
    d-cinema Package, 126  
    Essence, 118  
    Packing List, 95  
    Track Files, 104  
DCP Integrity, 126  
Decoder Requirements, 202  
Default Timed Text Font, 213  
Digital Audio Interfaces, 204

## E

Electronic Marriage Break Key Retaining, 223  
Entity Name and Roles Field, 25  
Equipment List  
    Hardware, 401  
    Software, 402  
ETM AnnotationText Language, 54  
ETM EncryptionMethod Element, 53  
ETM IssueDate Field Check, 84  
ETM ReferenceList Element, 55  
ETM Signature DigestMethod Element, 60  
ETM Signature Reference Elements, 57  
ETM Signature Transforms Field, 59  
ETM Signature Validity, 61  
ETM SignatureMethod Element, 58  
ETM SignedInfo CanonicalizationMethod Element, 56

Version 1.0 ARCHIVE ONLY  
NEWER VERSION AVAILABLE  
Version 1.1 released May 8 2009

ETM Signer Element, 52  
ETM Structure, 50  
ETM Validity Date Check, 51  
Event Log Operations, 153  
Event Logs  
    Logging, 149

## F

FIPS Testing  
    Type 1 SPB, 269  
FM Algorithm General Requirements, 290  
FM Application Constraints, 198  
FM Control Instance, 291  
FM Generic Inserter Requirements, 289  
FM Insertion Requirements, 290  
FM Payload, 200  
Forensic Marking, 198  
FrameSequencePlayed Event, 163

## G

General Log System Failure, 157  
GPIO Test Fixture  
    GPIO Test Fixture, 435  
Granularity of FM Control, 199  
Grayscale Tracking, 244

## I

Idempotency of ITM RRP, 287  
IFM Robustness, 290  
IFM Visual Transparency, 290  
Image and Audio Packaging Standard, 109  
Image Compression Standard & Encoding Parameters, 120  
Image Integrity Checking, 181  
Image Keys Outside Secure Silicon, 293  
Image Reproduction, 201  
Image Structure Container and Image Container Format, 118  
Image Track File Frame Boundary, 114  
IMB Link Encryption, 194  
IMB Log Storage Capabilities, 292  
IMB Tasks, 293  
Implementation of Proprietary ITMs, 288  
Implementation of RRP, 287  
Ingest and Storage, 249  
Interrupt Free Playback, 259  
Intra-frame Contrast, 243  
Intra-Theater Communication, 132  
Issuer Certificate Presence Check, 40

## K

KDM CipherData CPL ID, 80  
KDM CipherData Signer Thumbprint, 78  
KDM CipherData Structure ID, 77  
KDM CipherData Validity, 79

KDM CompositionPlaylistId Element, 74  
KDM ContentAuthenticator Element, 64  
KDM ContentTitleText Language Attribute, 74  
KDM DeviceListDescription Element, 70  
KDM EncryptedData Element, 68  
KDM EncryptedKey KeyType, 81  
KDM EncryptionMethod, 73  
KDM ForensicMarkFlagList Element, 67  
KDM KeyIdList Element, 76  
KDM KeyIdList/TypedKeyId Field, 66  
KDM KeyType Scope Attribute, 72  
KDM MessageType Element, 62  
KDM NonCriticalExtensions Element, 83  
KDM Purge upon Expiry, 284  
KDM Recipient X509IssuerName, 82  
KDM Signer Certificate Presence, 65  
KDM SubjectName Element, 63  
KDM Validity Checks, 257  
KDM Validity Fields, 75  
KDMDeleted Event, 169  
KDMKeysReceived Event, 168  
Key Delivery Message  
    Example, 45  
Key Usage Time Window, 284  
KeyInfo Element, 69  
KeyInfo Field Check, 89  
KeyTransfer Event, 174  
KeyUsage Field, 19

## L

LDB ASM Conformity, 231  
LDB Key Purging, 233  
LDB Key Storage, 232  
LDB Logging, 235  
LDB Time-Awareness, 230  
LDB TLS Session Constraints, 229  
LDB Trust, 190  
LDB without Electronic Marriage, 228  
LE Key Generation, 274  
LE Key Issuance, 284  
LE Key Usage, 193  
Link Decryptor Block, 228  
Link Encryption, 190  
LinkClosed Event, 171  
LinkException Event, 172  
LinkOpened Event, 170  
Location of Security Manager, 282  
Log Collection by the SM, 156  
Log Event Proxy  
    Log Proxy, 158  
Log Purging in failed SPBs, 293  
Log Records for Multiple SPBs, 154  
Log Sequence Numbers, 155



Log Structure, 153  
Logging for Standalone Systems, 292  
Logging of Failed Procedures, 292  
LogTransfer Event, 173

## M

Maximum Key Validity Period, 284  
Maximum Number of DCP Keys, 85  
MD Caching of Keys, 295  
Media Block Overlay, 210  
Missing Required Fields, 31  
Multiple LE Operation, 191

## O

Organization Name Field, 23  
OrganizationName Match Check, 35  
OrganizationUnitName Field, 24  
Overview, 1

## P

Packing List File, 96  
Packing List Signature Validation, 98  
PathLen Check, 33  
Playback of Image Only Material, 201  
Playback Preparation, 283  
PlayoutComplete Event, 166  
Preloaded Symmetric Key Entropy, 295  
Prevention of Keying of Compromised SPBs, 283  
Private Keys outside Secure Silicon, 293  
Prohibition of SPB1 Field Serviceability, 294  
Projector  
    Consolidated Test Sequence, 313, 319  
    Test Environment, 215  
Projector Access Door, 217  
Projector Companion SPB Location, 224  
Projector Image Reproduction, 236  
Projector Lens, 237  
Projector Overlay, 236  
Projector Physical Protection, 216  
Projector Pixel Count/Structure, 238  
Projector Secure Silicon Device, 285  
Projector Spatial Resolution and Frame Rate Conversion, 239  
Public Key Thumbprint, 21  
Public Key Type Check, 39

## R

Reference Documents, 437  
Remote SPB Integrity Monitoring, 187  
Remote SPB Log Storage Requirements, 292  
Repair and Renewal of SPBs, 286  
Restarting Playback, 261  
Restriction of Keying to MD Type, 185  
Restriction of Keying to Monitored Link Decryptors, 184

Restriction of Keying to valid CPLs, 186  
RRP "Busy" and unsupported Types, 289  
RRP Broadcast Prohibition, 288  
RRP Initiator, 288  
RRP Operational Message Ports, 289  
RRP Synchronism, 287  
RSA Key Entropy, 294  
RSA Key Format, 16

## S

Screen Management System, 253, 253  
SE Log Authoring, 291  
SE Time Stamping, 291  
SE Uniqueness Constraint, 283  
Secure Processing Block, 129  
Secure SMS-SM Communication, 282  
Security Design Description Requirements, 275  
Security Devices Self-Test Capabilities, 281  
Security Entity Physical Protection, 281  
Security Log Events  
    ASM, 170  
    Log Events, 163  
Security Manager, 181  
SecurityLog Events  
    Playout, 163  
Sequential Contrast, 242  
SerialNumber Field, 14  
Server  
    Consolidated Test Sequence, 305  
Show Playlist Creation, 254  
Show Playlist Format, 256  
Signature Algorithm Check, 38  
Signature Validation, 27  
SignatureAlgorithm Fields, 12  
SignatureValue Field, 13  
SM Operating Environment, 274  
SM Proxy of Log Events, 158  
SM Proxy of Security ASM Events, 161  
SM Proxy of Security Operations Events, 159  
SM Secure Communications, 282  
SM TLS Session Partners, 289  
SM Usage of OS Security Features, 282  
SMS and SPB Authentication and ITM Transport Layer, 287  
SMS Identity and Certificate, 264  
SMS Operator Identification, 263  
SMS User Accounts, 262  
Sound Integrity Checking, 183  
Source Code, 405  
SPB 1 Firmware Modifications, 276  
SPB Authentication, 283  
SPB Digital Certificate, 129  
SPB Integrity Fault Consequences, 189  
SPB Log Failure, 293

SPB Log Storage Requirements, 292  
SPB TLS Session Partners, 288  
SPB Type 2, 216  
SPB Type 2 Secure Silicon, 131  
SPB Type 2 Security Perimeter, 130  
SPB1 FIPS Requirements, 275  
SPB1 Secure Silicon FIPS Requirements, 275  
SPB1 Tamper Resistance, 275  
SPB1 Tamper Responsiveness, 274  
SPB2 Protected Devices, 286  
SPB2 Requirements, 218  
SPB2 Secure Silicon Field Replacement, 221  
SPB2 Secure Silicon Requirements, 219  
SPB2 Tamper Evidence, 220  
SPBClockadjust Event, 177  
SPBMarriage and SPBDivorce Events, 178  
SPBOpen SPBClose Events, 176  
SPBSoftware Event, 179  
SPBStartup SPBShutdown Events, 175  
Storage System Capacity, 250  
Storage System Ingest Interface, 249  
Storage System Performance, 252  
Storage System Redundancy, 251  
Structure ID Check, 86  
SubjectPublicKeyInfo Field, 15  
Support for Multiple Captions, 212  
Systems with Electronic Marriage, 285  
Systems Without Electronic Marriage, 222, 285

## T

Test Materials, 329  
Theater System Reliability, 281  
Theater System Storage Security, 281  
Timed Text Reproduction, 210  
Timed Text Resource Encoding, 124  
Timed Text Synchronization, 211  
Timed Text Track File Format, 111  
TLS Endpoints, 287  
TLS Mode Bypass Prohibition, 288  
TLS Session Initiation, 132  
TLS Session Key Refreshes, 283  
TMS role, 294  
Track File Key Epoch, 103  
Track File Length, 113  
Transfer Function, 246

## U

Unrecognized Extensions, 26  
Use of Software Protection Methods, 294

## V

Validity Date Check, 37  
Validity Field, 17

Volume Index File, 94

## W

White Point Chromaticity and Uniformity, 241  
White Point Luminance and Uniformity, 240

## X

XML tutorial, 41