

FILE COMPARISON

Produced: 2008-6-19 16:23:26

Mode: All Lines

Left file: D:\Works\TTCN-3\OMA相关文档\TTCN-3\OMA-IOP-TTCN-2008-Prague\AtsBCast_FileAndStreamDistributionTests-new.ttcn

Right file: D:\Works\TTCN-3\OMA相关文档\TTCN-3\OMA-IOP-TTCN-2008-Prague\AtsBCast_FileAndStreamDistributionTests.ttcn

```

1/**
2 *
3 * @author
4 *   ETSI CTI BCAST CON Project
5 * @version
6 *   $Id$
7 * @desc
8 *   This module specifies pilot tests for BCast
file and stream distribution.
9 */
10module AtsBCast_FileAndStreamDistributionTests {
11    import from LibBCast_Interface all;
12    import from AtsBCast_Main_Functions all;
13    import from
LibBCast_ServiceGuide_Templates all;
14    import from
AtsBCast_ServiceGuide_Functions all;
15    import from
LibBCast_ServiceGuide_TypesAndValues all;
16    import from LibBCast_Common_TypesAndValues
all;
17    import from LibCommon_BasicTypesAndValues
all;
18    import from AtsBCast_ModuleParameters all;
19    import from LibBCast_ModuleParameters all;
20    import from
AtsBCast_TestConfiguration_Functions all;
21    import from AtsBCast_TestSystem all;
22    // change 2 (WK14): file distribution as
well as preview data
23    import from
LibBCast_ServicePrimitives_TypesAndValues all;
24
25    group bcastConformanceTestCases {
26        group clientConformanceTestCases {
27            /**

```

```

=
1/**
2 *
3 * @author
4 *   ETSI CTI BCAST CON Project
5 * @version
6 *   $Id$
7 * @desc
8 *   This module specifies pilot tests for BCast
file and stream distribution.
9 */
10module AtsBCast_FileAndStreamDistributionTests {
11    import from LibBCast_Interface all;
12    import from AtsBCast_Main_Functions all;
13    import from
LibBCast_ServiceGuide_Templates all;
14    import from
AtsBCast_ServiceGuide_Functions all;
15    import from
LibBCast_ServiceGuide_TypesAndValues all;
16    import from LibBCast_Common_TypesAndValues
all;
17    import from LibCommon_BasicTypesAndValues
all;
18    import from AtsBCast_ModuleParameters all;
19    import from LibBCast_ModuleParameters all;
20    import from
AtsBCast_TestConfiguration_Functions all;
21    import from AtsBCast_TestSystem all;
22    // change 2 (WK14): file distribution as
well as preview data
23    import from
LibBCast_ServicePrimitives_TypesAndValues all;
24
25    group bcastConformanceTestCases {
26        group clientConformanceTestCases {
27            /**

```

28	*		28	*
29	* @desc		29	* @desc
30	* <p>Support of ALC	+ -		
	protocol and delivery of meta-data in the Service			
	Guide</p>			
31	* <p>Test Case			
	Description</p>			
32	* <p>with a terminal			
	having received Service Guide information related			
	to a file ready for download via broadcast channel			
	using ALC.</p>			
33	* <p>when the terminal			
	downloads file via broadcast channel</p>			
34	* <p>then the terminal			
	receives file and is able to display it </p>			
35	* <p>Specification			
	Reference</p>			
36	* <p>[BCAST10-			
	Distribution] Section 5.2, 5.1.2.5.3.2</p>			
37	* <p>Preconditions</p>			
38	* <p>Set up the test			
	tool for Service Guide delivery to provide file			
	delivery information for "File1". "File1" is			
	available on the broadcast channel.</p>			
39	* <p>This test cases			
	uses the following SG fragment instantiations			
	found in the Appendix:			
40	* 			
41	* Service			
	fragment with Name="FILE"			
42	* Content			
	fragment with Name="File1"			
43	* Schedule			
	fragment with contentLocation set to URI for "name-			
	of-the-file1".			
44	* Access			
	fragment with BroadcastServiceDelivery element			
	containing 			
45	* SessionDescription for an ALC session			
	indicating FEC encoding 0 (Compact No-Code FEC)			
	and FileDescription with attributes Content-			
	Location="name-of-the-file1" and TOI="1".			

```
46      *      <li>Transport
Object with TOI="1".</li>
47      *      </ul>
48      *      <p>Note:
49      *      <ul>
50      *      <li>All
the fragments are associated with the same Service
fragment.</li>
51      *      <li>IPv4
is used</li>
52      *      <li>File
is GZIP encoded</li>
53      *      </ul>
54      *      </p>
55      *      </p>
56      *      <p>Test Procedure</p>
57      *      <p>
58      *      <ul>
59      *      <li>Setup the
test tool to distribute file "File1" via the
broadcast channel</li>
60      *      <li>Set up
the terminal to receive BCAST service guide
announcements</li>
61      *      <li>Browse
the SG in the terminal and select the "File1" to
download</li>
62      *      <li>Wait for
the file download</li>
63      *      </ul>
64      *      </p>
65      *      <p>Note: "File1"
can be a jpg picture</p>
66      *      <p>Pass-Criteria</p>
67      *      <p>The following
should be visible to the end user
68      *      <ul>
69      *      <li>There is
a service "FILE" associated with file "File1"</li>
70      *      <li>The file
is successfully downloaded to the terminal.</li>
71      *      </ul>
72      *      </p>
```

```

73      *      <p>Note: To verify
the file was correctly downloaded the picture
should be correctly displayed.</p>
74      * @verdict
75      *      pass in case the
client pass the conformance test.
76      * @verdict
77      *      fail in case the
client does not pass the conformance test.
78      * @verdict
79      *      inconc in case a
guard timer expires.
80      */
81      testcase
TC_BCAST_DIST_CONF_101() runs on
BCastMainComponent system BCastPlatformComponent {
82          // init components
and ports
83          f_createComponents
();
84          f_cf_Broadcast_up
();
85          f_cf_Data_up();
86          f_cf_UpperTester_up
();
87
88          // preamble
89          var UInt
v_TransferId := 1;
90
f_main_data_startFileTransfer(v_TransferId,
PX_SDP_FILE_DELIVERY, {{fileRef :=
PX_PICTURE_ID}});
91
f_startCommonUtPreamble();
92
93          // test behavior
94          var
ServiceGuideDeliveryUnit v_SGDU := {};
95          var UInt32
v_FragmentVersion := 1;
96
97          var UInt
v_NTP_StartTime := 0;

```

```
98      var UInt
v_NTP_EndTime := 0;
99      f_InitStartEndTime
(v_NTP_StartTime, v_NTP_EndTime);
100
101      var DateTime
v_DateTime_StartTime := f_getDateTime
(v_NTP_StartTime);
102      var DateTime
v_DateTime_EndTime := f_getDateTime(v_NTP_EndTime);
103
104      var SdpFragment
v_Sdp := valueof(m_def_SdpFragment
(PX_SDP_VIDEO_PROG_1_REF_ID,
PX_SDP_FILE_DELIVERY));
105      var AccessTypeType
v_AccessType := valueof(m_def_AccessType_bc_sdp
(f_getAccessType(PX_BROADCAST_NETWORK_BEARER),
PX_SDP_VIDEO_PROG_1_REF_ID));
106
107      var ServiceType
v_Service := valueof(m_def_Service(
108          "service/
file",
109      v_FragmentVersion,
110          "FILE1",
111      c_basicTv_ServiceType
112          ));
113
114      var ContentType
v_Content := valueof(m_def_Content(
115          "content/
file1",
116      v_FragmentVersion,
117          "File1",
118          "service/
file",
119      v_DateTime_StartTime,
120      v_DateTime_EndTime
```

```
121                ));
122
123                var ScheduleType
v_Schedule := valueof(m_def_Schedule(
124    PX_SGDU_SCHEDULE_ID_PROG_1,
125    v_FragmentVersion,
126                                "service/
file",
127                                "content/
file1",
128    v_NTP_StartTime,
129    v_NTP_EndTime
130                ));
131
132                var AccessType
v_Access := valueof(m_def_Access(
133                                "access/
file1",
134    v_FragmentVersion,
135    v_AccessType,
136                                "service/
file",
137    PX_SGDU_SCHEDULE_ID_PROG_1,
138                                c_class_SG
139                ));
140
141    f_addServiceFragment(v_SGDU, v_Service);
142    f_addContentFragment(v_SGDU, v_Content);
143    f_addScheduleFragment(v_SGDU, v_Schedule);
144                                f_addAccessFragment
(v_SGDU, v_Access);
145                                f_addSdpFragment
(v_SGDU, v_FragmentVersion, v_Sdp);
146
```

147	f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);			
148				
149	f_main_ut_RunBCastApplication();			
150	if			
	(PX_GET_SG_DISPLAY_MANUALLY) {			
151	f_main_ut_GetServiceGuide(PX_SGDD_ID);			
152	}			
153				
	f_main_ut_CheckService("FILE");			
154	f_main_ut_SelectService("FILE");			
155	f_main_ut_CheckFile			
	("File1");			
156				
	f_main_ut_SelectFile("File1");			
157				
	f_main_ut_CheckFileReceived("Picture");			
158				
159	// postamble			
160				
	f_main_data_stopFileTransfer(v_TransferId);			
161				
162	f_cf_Broadcast_down			
	();			
163	f_cf_Data_down();			
164				
	f_cf_UpperTester_down();			
165	}// end test case			
	TC_BCAST_DIST_CONF_101			
166	/**			
167	*			
168	* @desc			
169	* <p>Support of in-	=	30	* <p>Support of in-
	band delivery of meta-data and FLUTE (optional)</p>			band delivery of meta-data and FLUTE (optional)</p>
170	* <p>Test Case		31	* <p>Test Case
	Description</p>			Description</p>
171	* <p>with a terminal		32	* <p>with a terminal
	having received Service Guide information related			having received Service Guide information related
	to a file ready for download via broadcast channel			to a file ready for download via broadcast channel
	using FLUTE.</p>			using FLUTE.</p>

172 * <p>when the terminal
downloads file via broadcast channel</p>
173 * <p>then the terminal
receives file and is able to display it </p>
174 * <p>Specification
Reference</p>
175 * <p>[BCAST10-
Distribution] Section 5.2.6, 5.1.2.5.3.1</p>
176 * <p>Preconditions</p>
177 * <p>Set up the test
tool for Service Guide delivery to provide file
delivery information for "File2". "File2" is
available on the broadcast channel.</p>
178 * <p>This test cases
uses the following SG fragment instantiations
found in the Appendix:
179 *
180 * Service
fragment with Name="FILE"
181 * Content
fragment with Name="File2"
182 * Schedule
fragment with contentLocation set to URI for "name-
of-the-file1".
183 * Access
fragment with BroadcastServiceDelivery element
containing
184 * SessionDescription for a FLUTE session
indicating FEC encoding 0 (Compact No-Code FEC)
and not containing FileDescription.
185 * Transport
Object with TOI="1".
186 *
187 * </p>
188 * <p>Set the FLUTE
session to contain an FDT-Instance with one File-
element having attributes Content-Location="name-
of-the-file1" and TOI="1"</p>
189 * <p>Note:
190 *
191 * All the
fragments are associated with the same Service
fragment.

33 * <p>when the terminal
downloads file via broadcast channel</p>
34 * <p>then the terminal
receives file and is able to display it </p>
35 * <p>Specification
Reference</p>
36 * <p>[BCAST10-
Distribution] Section 5.2.6, 5.1.2.5.3.1</p>
37 * <p>Preconditions</p>
38 * <p>Set up the test
tool for Service Guide delivery to provide file
delivery information for "File2". "File2" is
available on the broadcast channel.</p>
39 * <p>This test cases
uses the following SG fragment instantiations
found in the Appendix:
40 *
41 * Service
fragment with Name="FILE"
42 * Content
fragment with Name="File2"
43 * Schedule
fragment with contentLocation set to URI for "name-
of-the-file1".
44 * Access
fragment with BroadcastServiceDelivery element
containing
45 * SessionDescription for a FLUTE session
indicating FEC encoding 0 (Compact No-Code FEC)
and not containing FileDescription.
46 * Transport
Object with TOI="1".
47 *
48 * </p>
49 * <p>Set the FLUTE
session to contain an FDT-Instance with one File-
element having attributes Content-Location="name-
of-the-file1" and TOI="1"</p>
50 * <p>Note:
51 *
52 * All the
fragments are associated with the same Service
fragment.


```

192      *      <li>IPv4 is
used</li>
193      *      <li>File is
GZIP encoded</li>
194      *      </ul>
195      *      </p>
196      *      <p>Test Procedure</p>
197      *      <p>
198      *      <ul>
199      *      <li>Setup the
test tool to distribute file "File2" via the
broadcast channel</li>
200      *      <li>Set up
the terminal to receive BCAST service guide
announcements</li>
201      *      <li>Browse
the SG in the terminal and select the "File2" to
download</li>
202      *      <li>Wait for
the file download</li>
203      *      </ul>
204      *      </p>
205      *      <p>Note:  "File2"
can be a jpg picture</p>
206      *      <p>Pass-Criteria</p>
207      *      <p>The following
should be visible to the end user
208      *      <ul>
209      *      <li>There is
a service "FILE" associated with file "File2"</li>
210      *      <li>The file
is successfully downloaded to the terminal.</li>
211      *      </ul>
212      *      </p>
213      *      <p>Note:  To verify
the file was correctly downloaded the picture
should be correctly displayed.</p>
214      *      @verdict
215      *      pass in case the
client pass the conformance test.
216      *      @verdict
217      *      fail in case the
client does not pass the conformance test.
218      *      @verdict

```

```

53      *      <li>IPv4 is
used</li>
54      *      <li>File is
GZIP encoded</li>
55      *      </ul>
56      *      </p>
57      *      <p>Test Procedure</p>
58      *      <p>
59      *      <ul>
60      *      <li>Setup the
test tool to distribute file "File2" via the
broadcast channel</li>
61      *      <li>Set up
the terminal to receive BCAST service guide
announcements</li>
62      *      <li>Browse
the SG in the terminal and select the "File2" to
download</li>
63      *      <li>Wait for
the file download</li>
64      *      </ul>
65      *      </p>
66      *      <p>Note:  "File2"
can be a jpg picture</p>
67      *      <p>Pass-Criteria</p>
68      *      <p>The following
should be visible to the end user
69      *      <ul>
70      *      <li>There is
a service "FILE" associated with file "File2"</li>
71      *      <li>The file
is successfully downloaded to the terminal.</li>
72      *      </ul>
73      *      </p>
74      *      <p>Note:  To verify
the file was correctly downloaded the picture
should be correctly displayed.</p>
75      *      @verdict
76      *      pass in case the
client pass the conformance test.
77      *      @verdict
78      *      fail in case the
client does not pass the conformance test.
79      *      @verdict

```

```

219      *      inconc in case a
guard timer expires.
220      */
221      testcase
TC_BCAST_DIST_CONF_102() runs on
BCastMainComponent system BCastPlatformComponent {
222          // init components
and ports
223          f_createComponents
();
224          f_cf_Broadcast_up
();
225          f_cf_Data_up();
226          f_cf_UpperTester_up
();
227
228          //
preamble
229      f_startCommonUtPreamble();
230
231          // test behavior
232      f_main_ctrl_InitStartTime(c_one_hour,
c_one_hour);          // change 01 (WK
6): global time definition
233
234          var DateTime
v_DateTime_StartTime := f_getDateTime
(v_startTime);          // change 01 (WK 6): global
time definition
235          var DateTime
v_DateTime_EndTime := f_getDateTime
(v_endTime);          // change 01 (WK 6):
global time definition
236
237          // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
file name shall be 'name-of-the-file1'
238          var charstring
distributedFile := "name-of-the-file1";
239
240          // change 2
(WK14): file distribution

```

```

80      *      inconc in case a
guard timer expires.
81      */
82      testcase
TC_BCAST_DIST_CONF_102() runs on
BCastMainComponent system BCastPlatformComponent {
83          // init components
and ports
84          f_createComponents
();
85          f_cf_Broadcast_up
();
86          f_cf_Data_up();
87          f_cf_UpperTester_up
();
88
89          //
preamble
90      f_startCommonUtPreamble();
91
92          // test behavior
93      f_main_ctrl_InitStartTime(c_one_hour,
c_one_hour);          // change 01 (WK
6): global time definition
94
95          var DateTime
v_DateTime_StartTime := f_getDateTime
(v_startTime);          // change 01 (WK 6): global
time definition
96          var DateTime
v_DateTime_EndTime := f_getDateTime
(v_endTime);          // change 01 (WK 6):
global time definition
97
98          // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
file name shall be 'name-of-the-file1'
99          var charstring
distributedFile := "name-of-the-file1";
100
101          // change 2
(WK14): file distribution

```

```

241 // change 34
(WK18): rename "TVChannelIcon" SDP session into
"Data_Session", rename respective SDP PIXIT
paramter as well
242 var
TransferFileList v_FileList := {
243 {
244 Content_Location := distributedFile, // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
file name shall be 'name-of-the-file1'
245 Content_Type := "image/jpeg",
246 Content-Encoding := "gzip", // change 30 (WK18):
according to TC BCAST-1.0-DIST-conf-102 ->
distributed files shall be GZIP encoded
247 file := omit // use the content-Location as file
reference
248 }
249 };
250 var UInt
v_TransferId := 4;
251 // change 34
(WK18): rename "TVChannelIcon" SDP session into
"Data_Session", rename respective SDP PIXIT
paramter as well
252 f_main_data_startFileTransfer(v_TransferId,
PX_SDP_DATA_SESSION, v_FileList);
253
254 var
ServiceGuideDeliveryUnit v_SGDU := {};
255 var UInt32
v_FragmentVersion := 1;
256
257 v_TransferId := 1;
258
259 // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
access fragment containing SDP

```

```

102 // change 34
(WK18): rename "TVChannelIcon" SDP session into
"Data_Session", rename respective SDP PIXIT
paramter as well
103 var
TransferFileList v_FileList := {
104 {
105 Content_Location := distributedFile, // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
file name shall be 'name-of-the-file1'
106 Content_Type := "image/jpeg",
107 Content-Encoding := "gzip", // change 30 (WK18):
according to TC BCAST-1.0-DIST-conf-102 ->
distributed files shall be GZIP encoded
108 file := omit // use the content-Location as file
reference
109 }
110 };
111 var UInt
v_TransferId := 4;
112 // change 34
(WK18): rename "TVChannelIcon" SDP session into
"Data_Session", rename respective SDP PIXIT
paramter as well
113 f_main_data_startFileTransfer(v_TransferId,
PX_SDP_DATA_SESSION, v_FileList);
114
115 var
ServiceGuideDeliveryUnit v_SGDU := {};
116 var UInt32
v_FragmentVersion := 1;
117
118 v_TransferId := 1;
119
120 // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
access fragment containing SDP

```

```

260      var AccessTypeType
v_AccessType := valueof(m_def_AccessType
(f_getAccessType(PX_BROADCAST_NETWORK_BEARER),
fx_charstring2Base64(PX_SDP_DATA_SESSION)));
261
262      var ServiceType
v_Service := valueof(m_def_Service(
263      PX_SGDU_SERVICE_ID,
264      v_FragmentVersion,
265      "FILE",
266      c_basicTv_ServiceType
267      ));
268
269      var ContentType
v_Content := valueof(m_def_Content(
270      PX_SGDU_CONTENT_ID_PROG_1,
271      v_FragmentVersion,
272      "File2",
273      PX_SGDU_SERVICE_ID,
274      v_DateTime_StartTime,
275      v_DateTime_EndTime
276      ));
277
278      var ScheduleType
v_Schedule := valueof(m_def_Schedule(
279      PX_SGDU_SCHEDULE_ID_PROG_1,
280      v_FragmentVersion,
281      PX_SGDU_SERVICE_ID,
282      PX_SGDU_CONTENT_ID_PROG_1,
283      v_startTime,
change 01 (WK 6): global time definition

```

//

```

121      var AccessTypeType
v_AccessType := valueof(m_def_AccessType
(f_getAccessType(PX_BROADCAST_NETWORK_BEARER),
fx_charstring2Base64(PX_SDP_DATA_SESSION)));
122
123      var ServiceType
v_Service := valueof(m_def_Service(
124      PX_SGDU_SERVICE_ID,
125      v_FragmentVersion,
126      "FILE",
127      c_basicTv_ServiceType
128      ));
129
130      var ContentType
v_Content := valueof(m_def_Content(
131      PX_SGDU_CONTENT_ID_PROG_1,
132      v_FragmentVersion,
133      "File2",
134      PX_SGDU_SERVICE_ID,
135      v_DateTime_StartTime,
136      v_DateTime_EndTime
137      ));
138
139      var ScheduleType
v_Schedule := valueof(m_def_Schedule(
140      PX_SGDU_SCHEDULE_ID_PROG_1,
141      v_FragmentVersion,
142      PX_SGDU_SERVICE_ID,
143      PX_SGDU_CONTENT_ID_PROG_1,
144      v_startTime,
change 01 (WK 6): global time definition

```

//

```

284 v_endTime //
change 01 (WK 6): global time definition
285     ));
286     // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
with content location to 'name-of-the-file1'
287     var integer i;
288     for (i := 0; i <
sizeof(v_Schedule.ContentReferences); i := i + 1) {
289         v_Schedule.
ContentReferences[i].contentLocation :=
distributedFile
290     }
291
292     var AccessType
v_Access := valueof(m_def_Access(
293     PX_SGDU_ACCESS_ID_PROG_1,
294     v_FragmentVersion,
295     v_AccessType,
296     PX_SGDU_SERVICE_ID,
297     PX_SGDU_SCHEDULE_ID_PROG_1, // change 3 (WK18):
in according to ETS; access fragment references to
schedule fragment
298     c_class_FC // change 35 (WK18): Check Service
Class of Access Fragments: ST, SG, FC: ST when
access references video, FC for file distribution,
otherwise SG.
299     ));
300
301     f_addServiceFragment(v_SGDU, v_Service);
302     f_addContentFragment(v_SGDU, v_Content);
303     f_addScheduleFragment(v_SGDU, v_Schedule);
304     f_addAccessFragment
(v_SGDU, v_Access);

```

```

145 v_endTime //
change 01 (WK 6): global time definition
146     ));
147     // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
with content location to 'name-of-the-file1'
148     var integer i;
149     for (i := 0; i <
sizeof(v_Schedule.ContentReferences); i := i + 1) {
150         v_Schedule.
ContentReferences[i].contentLocation :=
distributedFile
151     }
152
153     var AccessType
v_Access := valueof(m_def_Access(
154     PX_SGDU_ACCESS_ID_PROG_1,
155     v_FragmentVersion,
156     v_AccessType,
157     PX_SGDU_SERVICE_ID,
158     PX_SGDU_SCHEDULE_ID_PROG_1, // change 3 (WK18):
in according to ETS; access fragment references to
schedule fragment
159     c_class_FC // change 35 (WK18): Check Service
Class of Access Fragments: ST, SG, FC: ST when
access references video, FC for file distribution,
otherwise SG.
160     ));
161
162     f_addServiceFragment(v_SGDU, v_Service);
163     f_addContentFragment(v_SGDU, v_Content);
164     f_addScheduleFragment(v_SGDU, v_Schedule);
165     f_addAccessFragment
(v_SGDU, v_Access);

```

```

305 //f_addSdpFragment
(v_SGDU, v_FragmentVersion, v_Sdp); // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
SDP is part of access fragment
306
307 f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
308
309 f_main_ut_RunBCastApplication();
310 if
(PX_GET_SG_DISPLAY_MANUALLY) {
311 f_main_ut_GetServiceGuide(PX_SGDD_ID);
312 }
313
f_main_ut_CheckService("FILE");
314
f_main_ut_SelectService("FILE");
315 f_main_ut_CheckFile
("File2");
316
f_main_ut_SelectFile("File2");
317
f_main_ut_CheckFileReceived("Picture");
318
319 // postamble
320
f_main_data_stopFileTransfer(v_TransferId);
321
322 f_cf_Broadcast_down
();
323 f_cf_Data_down();
324
f_cf_UpperTester_down();
325 } // end test case
TC_BCAST_DIST_CONF_102
326 // change 39 (WK18):
testcase description extension
327 with {extension
"Description: 5.3.1.2 Support of in-band delivery
of meta-data and FLUTE (optional)"}
328 } // end group
clientConformanceTestCases

```

```

166 //f_addSdpFragment
(v_SGDU, v_FragmentVersion, v_Sdp); // change 36
(WK18): according to TC BCAST-1.0-DIST-conf-102 ->
SDP is part of access fragment
167
168 f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
169
170 f_main_ut_RunBCastApplication();
171 if
(PX_GET_SG_DISPLAY_MANUALLY) {
172 f_main_ut_GetServiceGuide(PX_SGDD_ID);
173 }
174
f_main_ut_CheckService("FILE");
175
f_main_ut_SelectService("FILE");
176 f_main_ut_CheckFile
("File2");
177
f_main_ut_SelectFile("File2");
178
f_main_ut_CheckFileReceived("Picture");
179
180 // postamble
181
f_main_data_stopFileTransfer(v_TransferId);
182
183 f_cf_Broadcast_down
();
184 f_cf_Data_down();
185
f_cf_UpperTester_down();
186 } // end test case
TC_BCAST_DIST_CONF_102
187 // change 39 (WK18):
testcase description extension
188 with {extension
"Description: 5.3.1.2 Support of in-band delivery
of meta-data and FLUTE (optional)"}
189 } // end group
clientConformanceTestCases

```

329	// end group bcastConformanceTestCases	190	// end group bcastConformanceTestCases
330}		191}	