

FILE COMPARISON

Produced: 2008-6-19 16:24:25

Mode: All Lines

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

Right file: D:\Works\TTCN-3\OMA相关文档\TTCN-3\OMA-IOP-TTCN-2008-Prague\AtsBCast_ServiceGuideTests.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 service guide.
9 */
10module AtsBCast_ServiceGuideTests {
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_UpperTesterPrimitives_TypesAndValues all;
17  import from LibBCast_Common_TypesAndValues all;
18  import from LibCommon_BasicTypesAndValues all;
19  import from AtsBCast_ModuleParameters all;
20  import from LibBCast_ModuleParameters all;
21  import from AtsBCast_TestConfiguration_Functions all;
22  import from AtsBCast_TestSystem all;
23  // change 2 (WK18): file distribution as well as preview data
24  import from LibBCast_ServicePrimitives_TypesAndValues all;
25
26  group bcastConformanceTestCases {
27    group clientConformanceTestCases {
28      /**
29       *
30       * @desc
31       *   <p>Service Guide update via broadcast channel
32      (same fragment id, higher version number)</p>
33       *   <p>Test Case Description</p>
34       *   <p>with a terminal having received a Service
35      Guide and listening to the broadcast Service Guide Delivery Channel.</p>
36       *   <p>when the terminal receives Service Guide
37      update containing a content fragment with same id and higher version</p>
38       *   <p>then the terminal presents the information of
39      the updated Service Guide</p>
40       *   <p>Specification Reference</p>
41       *   <p>[BCAST10-ESG] Section 5.4.2.1.2, 5.5 3rd
42      bullet item</p>
43       *   <p>Preconditions</p>
44       *   <p>Service guide cache of terminal is erased</p>
45       *   <p>Terminal is configured to listen to BCAST
46      service guide announcements and delivery on the broadcast channel</p>
47       *   <p>Access point information for service guide
48      entry point is configured in test tool</p>
49       *   <p>This test case uses the following SG fragment
50      instantiations found in the Appendix:</p>
51       *   <p>Initial service guide:
52       *   <ul>
53       *     <li>Service fragment with
54      Name="TvChannel"</li>
55       *     <li>Content fragment with version = "1"
56      and Name = "Programmel" , and StartTime and EndTime elements indicating values
57      after the time of test</li>
```

```
=
1/**
2 *
3 * @author
4 *   ETSI CTI BCAST CON Project
5 * @version
6 *   $Id$
7 * @desc
8 *   This module specifies pilot tests for BCast service guide.
9 */
10module AtsBCast_ServiceGuideTests {
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_UpperTesterPrimitives_TypesAndValues all;
17  import from LibBCast_Common_TypesAndValues all;
18  import from LibCommon_BasicTypesAndValues all;
19  import from AtsBCast_ModuleParameters all;
20  import from LibBCast_ModuleParameters all;
21  import from AtsBCast_TestConfiguration_Functions all;
22  import from AtsBCast_TestSystem all;
23  // change 2 (WK18): file distribution as well as preview data
24  import from LibBCast_ServicePrimitives_TypesAndValues all;
25
26  group bcastConformanceTestCases {
27    group clientConformanceTestCases {
28      /**
29       *
30       * @desc
31       *   <p>Service Guide update via broadcast channel
32      (same fragment id, higher version number)</p>
33       *   <p>Test Case Description</p>
34       *   <p>with a terminal having received a Service
35      Guide and listening to the broadcast Service Guide Delivery Channel.</p>
36       *   <p>when the terminal receives Service Guide
37      update containing a content fragment with same id and higher version</p>
38       *   <p>then the terminal presents the information of
39      the updated Service Guide</p>
40       *   <p>Specification Reference</p>
41       *   <p>[BCAST10-ESG] Section 5.4.2.1.2, 5.5 3rd
42      bullet item</p>
43       *   <p>Preconditions</p>
44       *   <p>Service guide cache of terminal is erased</p>
45       *   <p>Terminal is configured to listen to BCAST
46      service guide announcements and delivery on the broadcast channel</p>
47       *   <p>Access point information for service guide
48      entry point is configured in test tool</p>
49       *   <p>This test case uses the following SG fragment
50      instantiations found in the Appendix:</p>
51       *   <p>Initial service guide:
52       *   <ul>
53       *     <li>Service fragment with
54      Name="TvChannel"</li>
55       *     <li>Content fragment with version = "1"
56      and Name = "Programmel" , and StartTime and EndTime elements indicating values
57      after the time of test</li>
```

```
47      *          <li>Schedule fragment for "Programmel"
content fragment with same values for startTime and endTime in the
presentationWindow element</li>
48      *          <li>Access fragment "Programmel" schedule
fragment</li>
49      *          </ul>
50      *          </p>
51      *          <p>Updated service guide:
52      *          <ul>
53      *          <li>Service fragment with
Name="TvChannel"</li>
54      *          <li>Content fragment with version = "2"
and Name = "Programmel-version2" , and StartTime and EndTime elements
indicating values after the time of test</li>
55      *          <li>Schedule fragment for "Programmel-
version2" content fragment with same values for startTime and endTime in the
presentationWindow element</li>
56      *          <li>Access fragment for "Programmel-
version2" schedule fragment</li>
57      *          </ul>
58      *          Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.
59      *          </p>
60      *          <p>Test Procedure</p>
61      *          <p>
62      *          <ul>
63      *          <li>Set up the test tool to produce
initial BCAST service guide announcement and delivery using broadcast channel.</
li>
64      *          <li>Activate the BCAST application of the
terminal</li>
65      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
66      *          <li>Browse the SG on the terminal</li>
67      *          <li>Set up the test tool to produce
updated BCAST service guide announcement and delivery using broadcast channel.</
li>
68      *          <li>Request from BCAST application on
terminal to update the service guide (optional).</li>
69      *          <li>Browse the SG on the terminal</li>
70      *          </ul>
71      *          </p>
72      *          <p>Pass-Criteria</p>
73      *          <p>The following should be visible to the end
user after the initial delivery of the SG
74      *          <ul>
75      *          <li>There is a service "TvChannel"
associated with a program "Programmel" as well as start and end time values
(There is a "TvChannel" that contains "Programmel" scheduled from startTime to
endTime)</li>
76      *          </ul>
77      *          </p>
78      *          <p>The following should be visible to the end
user after the delivery of the update of the SG
79      *          <ul>
80      *          <li>There is a service "TvChannel"
associated with a program "Programmel-version2" as well as start and end time
values (There is a "TvChannel" that contains "Programmel-version2" scheduled
from startTime to endTime)</li>
81      *          </ul>
82      *          </p>
83      *          @verdict
84      *          pass in case the client pass the conformance test.
85      *          @verdict
86      *          fail in case the client does not pass the
conformance test.
```

```
47      *          <li>Schedule fragment for "Programmel"
content fragment with same values for startTime and endTime in the
presentationWindow element</li>
48      *          <li>Access fragment "Programmel" schedule
fragment</li>
49      *          </ul>
50      *          </p>
51      *          <p>Updated service guide:
52      *          <ul>
53      *          <li>Service fragment with
Name="TvChannel"</li>
54      *          <li>Content fragment with version = "2"
and Name = "Programmel-version2" , and StartTime and EndTime elements
indicating values after the time of test</li>
55      *          <li>Schedule fragment for "Programmel-
version2" content fragment with same values for startTime and endTime in the
presentationWindow element</li>
56      *          <li>Access fragment for "Programmel-
version2" schedule fragment</li>
57      *          </ul>
58      *          Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.
59      *          </p>
60      *          <p>Test Procedure</p>
61      *          <p>
62      *          <ul>
63      *          <li>Set up the test tool to produce
initial BCAST service guide announcement and delivery using broadcast channel.</
li>
64      *          <li>Activate the BCAST application of the
terminal</li>
65      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
66      *          <li>Browse the SG on the terminal</li>
67      *          <li>Set up the test tool to produce
updated BCAST service guide announcement and delivery using broadcast channel.</
li>
68      *          <li>Request from BCAST application on
terminal to update the service guide (optional).</li>
69      *          <li>Browse the SG on the terminal</li>
70      *          </ul>
71      *          </p>
72      *          <p>Pass-Criteria</p>
73      *          <p>The following should be visible to the end
user after the initial delivery of the SG
74      *          <ul>
75      *          <li>There is a service "TvChannel"
associated with a program "Programmel" as well as start and end time values
(There is a "TvChannel" that contains "Programmel" scheduled from startTime to
endTime)</li>
76      *          </ul>
77      *          </p>
78      *          <p>The following should be visible to the end
user after the delivery of the update of the SG
79      *          <ul>
80      *          <li>There is a service "TvChannel"
associated with a program "Programmel-version2" as well as start and end time
values (There is a "TvChannel" that contains "Programmel-version2" scheduled
from startTime to endTime)</li>
81      *          </ul>
82      *          </p>
83      *          @verdict
84      *          pass in case the client pass the conformance test.
85      *          @verdict
86      *          fail in case the client does not pass the
conformance test.
```

```

87      * @verdict
88      *   inconc in case a guard timer expires.
89      */
90      testcase TC_BCAST_ESG_CONF_101() runs on
BCastMainComponent system BCastPlatformComponent {
91          // init components and ports
92          f_createComponents();
93          f_cf_Broadcast_up();
94          f_cf_UpperTester_up();
95
96          // preamble
97          f_startCommonUtPreamble();
98
99          var ServiceGuideDeliveryUnit v_SGDU := {};
100          var UInt32 v_FragmentVersion := 1;
101
102          f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour);          // change 01 (WK 6): global time definition
103
104          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime);          // change 01 (WK 6): global time
definition
105          var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime);          // change 01 (WK 6): global time definition
106
107          var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
108              PX_SDP_VIDEO_PROG_1_REF_ID,
109              PX_SDP_VIDEO_PROG_1
110          ));
111
112          var AccessTypeType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
113              f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
114              PX_SDP_VIDEO_PROG_1_REF_ID
115          ));
116
117          var ServiceType v_Service := valueof
(m_def_Service(
118              PX_SGDU_SERVICE_ID,
119              v_FragmentVersion,
120              "TvChannel",
121              c_basicTv_ServiceType
122          ));
123
124          var ContentType v_Content := valueof
(m_def_Content(
125              PX_SGDU_CONTENT_ID_PROG_1,
126              v_FragmentVersion,
127              "Programm1",
128              PX_SGDU_SERVICE_ID,
129              v_DateTime_StartTime,
130              v_DateTime_EndTime
131          ));
132
133          var ScheduleType v_Schedule := valueof
(m_def_Schedule(
134              PX_SGDU_SCHEDULE_ID_PROG_1,
135              v_FragmentVersion,
136              PX_SGDU_SERVICE_ID,
137              PX_SGDU_CONTENT_ID_PROG_1,
138              v_startTime,          // change
01 (WK 6): global time definition

```

```

87      * @verdict
88      *   inconc in case a guard timer expires.
89      */
90      testcase TC_BCAST_ESG_CONF_101() runs on
BCastMainComponent system BCastPlatformComponent {
91          // init components and ports
92          f_createComponents();
93          f_cf_Broadcast_up();
94          f_cf_UpperTester_up();
95
96          // preamble
97          f_startCommonUtPreamble();
98
99          var ServiceGuideDeliveryUnit v_SGDU := {};
100          var UInt32 v_FragmentVersion := 1;
101
102          f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour);          // change 01 (WK 6): global time definition
103
104          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime);          // change 01 (WK 6): global time
definition
105          var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime);          // change 01 (WK 6): global time definition
106
107          var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
108              PX_SDP_VIDEO_PROG_1_REF_ID,
109              PX_SDP_VIDEO_PROG_1
110          ));
111
112          var AccessTypeType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
113              f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
114              PX_SDP_VIDEO_PROG_1_REF_ID
115          ));
116
117          var ServiceType v_Service := valueof
(m_def_Service(
118              PX_SGDU_SERVICE_ID,
119              v_FragmentVersion,
120              "TvChannel",
121              c_basicTv_ServiceType
122          ));
123
124          var ContentType v_Content := valueof
(m_def_Content(
125              PX_SGDU_CONTENT_ID_PROG_1,
126              v_FragmentVersion,
127              "Programm1",
128              PX_SGDU_SERVICE_ID,
129              v_DateTime_StartTime,
130              v_DateTime_EndTime
131          ));
132
133          var ScheduleType v_Schedule := valueof
(m_def_Schedule(
134              PX_SGDU_SCHEDULE_ID_PROG_1,
135              v_FragmentVersion,
136              PX_SGDU_SERVICE_ID,
137              PX_SGDU_CONTENT_ID_PROG_1,
138              v_startTime,          // change
01 (WK 6): global time definition

```

```

139 v_endTime                                     // change
140 01 (WK 6): global time definition
141     ));
142     var AccessType v_Access := valueof(m_def_Access(
143         PX_SGDU_ACCESS_ID_PROG_1,
144         v_FragmentVersion,
145         v_AccessType,
146         PX_SGDU_SERVICE_ID,
147         PX_SGDU_SCHEDULE_ID_PROG_1,
148         c_class_SG
149     ));
150
151     f_addServiceFragment(v_SGDU, v_Service);
152     f_addContentFragment(v_SGDU, v_Content);
153     f_addScheduleFragment(v_SGDU, v_Schedule);
154     f_addAccessFragment(v_SGDU, v_Access);
155     f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp);
156
157     f_main_ut_RunBCastApplication();
158     if (PX_GET_SG_DISPLAY_MANUALLY) {
159         f_main_ut_GetServiceGuide(PX_SGDD_ID);
160     }
161     f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
162     f_main_ut_CheckService("TvChannel");
163     f_main_ut_SelectService("TvChannel");
164     f_main_ut_CheckContent("Programm1",
v_DateTime_StartTime, v_DateTime_EndTime);
165
166     // test behavior
167
168     // reset TransportID of BCAST SG
169     vc_BCASTTransportID :=
1;
// change 07 (WK 6): in order to send
same TransportID
170     v_SGDU := {};
171     v_Content.version := f_IncFragmentVersion
(v_Content.version);
172     v_Content.Names := {{omit,"Programm1-
version2"}};
173     f_addServiceFragment(v_SGDU,
v_Service);
// change 08 (WK 6): SGDU must contian all
fragments
174     f_addContentFragment(v_SGDU, v_Content);
175     f_addScheduleFragment(v_SGDU,
v_Schedule);
// change 08 (WK 6): SGDU must contian all
fragments
176     f_addAccessFragment(v_SGDU,
v_Access);
// change 08 (WK 6): SGDU must contian all
fragments
177     f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp);
// change 08 (WK 6): SGDU must contian all fragments
178
179     f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
180
181     if (PX_GET_SG_UPDATE_MANUALLY) {
182         f_main_ut_UpdateServiceGuide();
183     }
184     f_main_ut_CheckService("TvChannel");
185     f_main_ut_SelectService("TvChannel");
186     f_main_ut_CheckContent("Programm1-version2",
v_DateTime_StartTime, v_DateTime_EndTime);
187
188

```

```

139 v_endTime                                     // change
140 01 (WK 6): global time definition
141     ));
142     var AccessType v_Access := valueof(m_def_Access(
143         PX_SGDU_ACCESS_ID_PROG_1,
144         v_FragmentVersion,
145         v_AccessType,
146         PX_SGDU_SERVICE_ID,
147         PX_SGDU_SCHEDULE_ID_PROG_1,
148         c_class_SG
149     ));
150
151     f_addServiceFragment(v_SGDU, v_Service);
152     f_addContentFragment(v_SGDU, v_Content);
153     f_addScheduleFragment(v_SGDU, v_Schedule);
154     f_addAccessFragment(v_SGDU, v_Access);
155     f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp);
156
157     f_main_ut_RunBCastApplication();
158     if (PX_GET_SG_DISPLAY_MANUALLY) {
159         f_main_ut_GetServiceGuide(PX_SGDD_ID);
160     }
161     f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
162     f_main_ut_CheckService("TvChannel");
163     f_main_ut_SelectService("TvChannel");
164     f_main_ut_CheckContent("Programm1",
v_DateTime_StartTime, v_DateTime_EndTime);
165
166     // test behavior
167
168     // reset TransportID of BCAST SG
169     vc_BCASTTransportID :=
1;
// change 07 (WK 6): in order to send
same TransportID
170     v_SGDU := {};
171     v_Content.version := f_IncFragmentVersion
(v_Content.version);
172     v_Content.Names := {{omit,"Programm1-
version2"}};
173     f_addServiceFragment(v_SGDU,
v_Service);
// change 08 (WK 6): SGDU must contian all
fragments
174     f_addContentFragment(v_SGDU, v_Content);
175     f_addScheduleFragment(v_SGDU,
v_Schedule);
// change 08 (WK 6): SGDU must contian all
fragments
176     f_addAccessFragment(v_SGDU,
v_Access);
// change 08 (WK 6): SGDU must contian all
fragments
177     f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp);
// change 08 (WK 6): SGDU must contian all fragments
178
179     f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
180
181     if (PX_GET_SG_UPDATE_MANUALLY) {
182         f_main_ut_UpdateServiceGuide();
183     }
184     f_main_ut_CheckService("TvChannel");
185     f_main_ut_SelectService("TvChannel");
186     f_main_ut_CheckContent("Programm1-version2",
v_DateTime_StartTime, v_DateTime_EndTime);
187
188

```

```
189 // postamble
190 f_main_ut_PowerOff();
191
192 f_cf_Broadcast_down();
193 f_cf_UpperTester_down();
194 } // end test case TC_BCAST_ESG_CONF_101
195 // change 39 (WK18): testcase description extension
196 with {extension "Description: 5.2.1 Service Guide
update via broadcast channel (same fragment id, higher version number)"}
197
198 /**
199 *
200 * @desc
201 * <p>Service Guide update via broadcast channel
(same fragment id, lower version number)</p>
202 * <p>Test Case Description</p>
203 * <p>with a terminal having received a Service
Guide and listening to the broadcast Service Guide Delivery Channel.</p>
204 * <p>when the terminal receives Service Guide
update containing a content fragment with same id and lower version</p>
205 * <p>then the terminal does not present the
information of the updated Service Guide</p>
206 * <p>Specification Reference</p>
207 * <p>[BCAST10-ESG] Section 5.4.2.1.2, 5.5 4th
bullet item</p>
208 * <p>Preconditions</p>
209 * <p>Service guide cache of terminal is erased</p>
210 * <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
211 * <p>Access point information for service guide
entry point is configured in test tool</p>
212 * <p>This test case uses the following SG fragment
instantiations found in the Appendix:</p>
213 * <p>Initial service guide:
214 * <ul>
215 * <li>Service fragment with
Name="TvChannel"</li>
216 * <li>Content fragment with version = "2"
and Name = "Programmel-version2" , and StartTime and EndTime elements
indicating values after the time of test</li>
217 * <li>Schedule fragment for "Programmel-
version2" content fragment with same values for startTime and endTime in the
presentationWindow element</li>
218 * <li>Access fragment for "Programmel-
version2" schedule fragment</li>
219 * </ul>
220 * </p>
221 * <p>Updated service guide:
222 * <ul>
223 * <li>Service fragment with
Name="TvChannel"</li>
224 * <li>Content fragment with version = "1"
and Name = "Programmel" , and StartTime and EndTime elements indicating values
after the time of test</li>
225 * <li>Schedule fragment for "Programmel"
content fragment with same values for startTime and endTime in the
presentationWindow element</li>
226 * <li>Access fragment for "Programmel"
schedule fragment</li>
227 * </ul>
228 * </p>
229 * <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery</p>
230 * <p>Test Procedure</p>
231 * <p>
232 * <ul>
```

```
189 // postamble
190 f_main_ut_PowerOff();
191
192 f_cf_Broadcast_down();
193 f_cf_UpperTester_down();
194 } // end test case TC_BCAST_ESG_CONF_101
195 // change 39 (WK18): testcase description extension
196 with {extension "Description: 5.2.1 Service Guide
update via broadcast channel (same fragment id, higher version number)"}
197
198 /**
199 *
200 * @desc
201 * <p>Service Guide update via broadcast channel
(same fragment id, lower version number)</p>
202 * <p>Test Case Description</p>
203 * <p>with a terminal having received a Service
Guide and listening to the broadcast Service Guide Delivery Channel.</p>
204 * <p>when the terminal receives Service Guide
update containing a content fragment with same id and lower version</p>
205 * <p>then the terminal does not present the
information of the updated Service Guide</p>
206 * <p>Specification Reference</p>
207 * <p>[BCAST10-ESG] Section 5.4.2.1.2, 5.5 4th
bullet item</p>
208 * <p>Preconditions</p>
209 * <p>Service guide cache of terminal is erased</p>
210 * <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
211 * <p>Access point information for service guide
entry point is configured in test tool</p>
212 * <p>This test case uses the following SG fragment
instantiations found in the Appendix:</p>
213 * <p>Initial service guide:
214 * <ul>
215 * <li>Service fragment with
Name="TvChannel"</li>
216 * <li>Content fragment with version = "2"
and Name = "Programmel-version2" , and StartTime and EndTime elements
indicating values after the time of test</li>
217 * <li>Schedule fragment for "Programmel-
version2" content fragment with same values for startTime and endTime in the
presentationWindow element</li>
218 * <li>Access fragment for "Programmel-
version2" schedule fragment</li>
219 * </ul>
220 * </p>
221 * <p>Updated service guide:
222 * <ul>
223 * <li>Service fragment with
Name="TvChannel"</li>
224 * <li>Content fragment with version = "1"
and Name = "Programmel" , and StartTime and EndTime elements indicating values
after the time of test</li>
225 * <li>Schedule fragment for "Programmel"
content fragment with same values for startTime and endTime in the
presentationWindow element</li>
226 * <li>Access fragment for "Programmel"
schedule fragment</li>
227 * </ul>
228 * </p>
229 * <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery</p>
230 * <p>Test Procedure</p>
231 * <p>
232 * <ul>
```

```

233      *          <li>Set up the test tool to produce
initial BCAST service guide announcement and delivery using broadcast channel.</
li>
234      *          <li>Activate the BCAST application of the
terminal</li>
235      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
236      *          <li>Browse the SG on the terminal</li>
237      *          <li>Set up the test tool to produce
updated BCAST service guide announcement and delivery using broadcast channel.</
li>
238      *          <li>Request from BCAST application on
terminal to update the service guide (optional).</li>
239      *          <li>Browse the SG on the terminal</li>
240      *          </ul>
241      *          </p>
242      *          <p>Pass-Criteria</p>
243      *          <p>The following should be visible to the end
user after the initial delivery of the SG
244      *          <ul>
245      *          <li>There is a service called "TvChannel"
associated with program "Programmel-version2" as well as start and end time
values(There is a "TvChannel" that contains "Programmel-version2" scheduled
from startTime to endTime</li>
246      *          </ul>
247      *          </p>
248      *          <p>The following should be visible to the end
user after the delivery of the updated SG
249      *          <ul>
250      *          <li>There is a service called "TvChannel"
associated with program "Programmel-version1" as well as start and end time
values(There is a "TvChannel" that contains "Programmel-version1" scheduled
from startTime to endTime)).</li>
251      *          </ul>
252      *          </p>
253      *          @verdict
254      *          pass in case the client pass the conformance test.
255      *          @verdict
256      *          fail in case the client does not pass the
conformance test.
257      *          @verdict
258      *          inconc in case a guard timer expires.
259      */
260      testcase TC_BCAST_ESG_CONF_102() runs on
BCastMainComponent system BCastPlatformComponent {
261          // init components and ports
262          f_createComponents();
263          f_cf_Broadcast_up
          ();
          // change 09 (WK
6): set up broadcast channel
          f_cf_UpperTester_up();
264
265          // preamble
266          f_startCommonUtPreamble();
267
268          var ServiceGuideDeliveryUnit v_SGDU := {};
269          var UInt32 v_FragmentVersion := 1;
270
271          f_main_ctrl_InitStartTime(c_one_hour,
272          c_one_hour);
          // change 01 (WK 6): global time
          definition
273
274          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime);
          // change 01 (WK 6): global time
          definition

```

```

233      *          <li>Set up the test tool to produce
initial BCAST service guide announcement and delivery using broadcast channel.</
li>
234      *          <li>Activate the BCAST application of the
terminal</li>
235      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
236      *          <li>Browse the SG on the terminal</li>
237      *          <li>Set up the test tool to produce
updated BCAST service guide announcement and delivery using broadcast channel.</
li>
238      *          <li>Request from BCAST application on
terminal to update the service guide (optional).</li>
239      *          <li>Browse the SG on the terminal</li>
240      *          </ul>
241      *          </p>
242      *          <p>Pass-Criteria</p>
243      *          <p>The following should be visible to the end
user after the initial delivery of the SG
244      *          <ul>
245      *          <li>There is a service called "TvChannel"
associated with program "Programmel-version2" as well as start and end time
values(There is a "TvChannel" that contains "Programmel-version2" scheduled
from startTime to endTime</li>
246      *          </ul>
247      *          </p>
248      *          <p>The following should be visible to the end
user after the delivery of the updated SG
249      *          <ul>
250      *          <li>There is a service called "TvChannel"
associated with program "Programmel-version1" as well as start and end time
values(There is a "TvChannel" that contains "Programmel-version1" scheduled
from startTime to endTime)).</li>
251      *          </ul>
252      *          </p>
253      *          @verdict
254      *          pass in case the client pass the conformance test.
255      *          @verdict
256      *          fail in case the client does not pass the
conformance test.
257      *          @verdict
258      *          inconc in case a guard timer expires.
259      */
260      testcase TC_BCAST_ESG_CONF_102() runs on
BCastMainComponent system BCastPlatformComponent {
261          // init components and ports
262          f_createComponents();
263          f_cf_Broadcast_up
          ();
          // change 09 (WK
6): set up broadcast channel
          f_cf_UpperTester_up();
264
265          // preamble
266          f_startCommonUtPreamble();
267
268          var ServiceGuideDeliveryUnit v_SGDU := {};
269          var UInt32 v_FragmentVersion := 1;
270
271          f_main_ctrl_InitStartTime(c_one_hour,
272          c_one_hour);
          // change 01 (WK 6): global time
          definition
273
274          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime);
          // change 01 (WK 6): global time
          definition

```

```

275      var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime);
// change 01 (WK 6): global time definition
276
277      var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
278          PX_SDP_VIDEO_PROG_1_REF_ID,
279          PX_SDP_VIDEO_PROG_1
280      ));
281
282      var AccessTypeType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
283          f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
284          PX_SDP_VIDEO_PROG_1_REF_ID
285      ));
286
287      var ServiceType v_Service := valueof
(m_def_Service(
288          PX_SGDU_SERVICE_ID,
289          v_FragmentVersion,
290          "TvChannel",
291          c_basicTv_ServiceType
292      ));
293
294      var ContentType v_Content := valueof
(m_def_Content(
295          PX_SGDU_CONTENT_ID_PROG_1,
296          f_IncFragmentVersion(v_FragmentVersion),
297          "Programm1-version2",
298          PX_SGDU_SERVICE_ID,
299          v_DateTime_StartTime,
300          v_DateTime_EndTime
301      ));
302
303      var ScheduleType v_Schedule := valueof
(m_def_Schedule(
304          PX_SGDU_SCHEDULE_ID_PROG_1,
305          v_FragmentVersion,
306          PX_SGDU_SERVICE_ID,
307          PX_SGDU_CONTENT_ID_PROG_1,
308          v_startTime,
309          v_endTime
310      ));
// change 01 (WK 6):
// change 01 (WK 6):
311
312      var AccessType v_Access := valueof(m_def_Access(
313          PX_SGDU_ACCESS_ID_PROG_1,
314          v_FragmentVersion,
315          v_AccessType,
316          PX_SGDU_SERVICE_ID,
317          PX_SGDU_SCHEDULE_ID_PROG_1,
318          c_class_SG
319      ));
320
321      f_addServiceFragment(v_SGDU, v_Service);
322      f_addContentFragment(v_SGDU, v_Content);
323      f_addScheduleFragment(v_SGDU, v_Schedule);
324      f_addAccessFragment(v_SGDU, v_Access);
325      f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp);
326
327      f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
328      f_main_ut_RunBCastApplication();

```

```

275      var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime);
// change 01 (WK 6): global time definition
276
277      var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
278          PX_SDP_VIDEO_PROG_1_REF_ID,
279          PX_SDP_VIDEO_PROG_1
280      ));
281
282      var AccessTypeType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
283          f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
284          PX_SDP_VIDEO_PROG_1_REF_ID
285      ));
286
287      var ServiceType v_Service := valueof
(m_def_Service(
288          PX_SGDU_SERVICE_ID,
289          v_FragmentVersion,
290          "TvChannel",
291          c_basicTv_ServiceType
292      ));
293
294      var ContentType v_Content := valueof
(m_def_Content(
295          PX_SGDU_CONTENT_ID_PROG_1,
296          f_IncFragmentVersion(v_FragmentVersion),
297          "Programm1-version2",
298          PX_SGDU_SERVICE_ID,
299          v_DateTime_StartTime,
300          v_DateTime_EndTime
301      ));
302
303      var ScheduleType v_Schedule := valueof
(m_def_Schedule(
304          PX_SGDU_SCHEDULE_ID_PROG_1,
305          v_FragmentVersion,
306          PX_SGDU_SERVICE_ID,
307          PX_SGDU_CONTENT_ID_PROG_1,
308          v_startTime,
309          v_endTime
310      ));
// change 01 (WK 6):
// change 01 (WK 6):
311
312      var AccessType v_Access := valueof(m_def_Access(
313          PX_SGDU_ACCESS_ID_PROG_1,
314          v_FragmentVersion,
315          v_AccessType,
316          PX_SGDU_SERVICE_ID,
317          PX_SGDU_SCHEDULE_ID_PROG_1,
318          c_class_SG
319      ));
320
321      f_addServiceFragment(v_SGDU, v_Service);
322      f_addContentFragment(v_SGDU, v_Content);
323      f_addScheduleFragment(v_SGDU, v_Schedule);
324      f_addAccessFragment(v_SGDU, v_Access);
325      f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp);
326
327      f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
328      f_main_ut_RunBCastApplication();

```

```

329         if (PX_GET_SG_DISPLAY_MANUALLY) {
330             f_main_ut_GetServiceGuide(PX_SGDD_ID);
331         }
332         f_main_ut_CheckService("TvChannel");
333         f_main_ut_SelectService("TvChannel");
334         f_main_ut_CheckContent("Programm1-version2",
v_DateTime_StartTime, v_DateTime_EndTime);
335
336         // test behavior
337         vc_BCASTTransportID :=
1; // change 07 (WK 6): in order to send
same TransportID
338         v_SGDU :=
{}; // change 10 (WK 6): empty
SGDU to avoid multiple content fragments
339         v_Content.version := v_FragmentVersion;
340         v_Content.Names := {{omit,"Programm1"}};
341         f_addServiceFragment(v_SGDU,
// change 08 (WK 6): SGDU must contain all
v_Service);
fragments
342         f_addContentFragment(v_SGDU, v_Content);
343         f_addScheduleFragment(v_SGDU,
// change 08 (WK 6): SGDU must contain all
v_Schedule);
fragments
344         f_addAccessFragment(v_SGDU,
// change 08 (WK 6): SGDU must contain all
v_Access);
fragments
345         f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp); // change 08 (WK 6): SGDU must contain all fragments
346
347         f_main_ctrl_broadcastServiceGuide(v_SGDU,
omit); // change 11 (WK 6): reset FLUTE; delivers new SGDU
348
349         if (PX_GET_SG_UPDATE_MANUALLY) {
350             f_main_ut_UpdateServiceGuide();
351         }
352         f_main_ut_CheckService("TvChannel");
353         f_main_ut_SelectService("TvChannel");
354         f_main_ut_CheckContent("Programm1-version2",
v_DateTime_StartTime, v_DateTime_EndTime);
355
356         // postamble
357         f_main_ut_PowerOff();
358
359         f_cf_Broadcast_down();
360         f_cf_UpperTester_down();
361     } // end test case TC_BCAST_ESG_CONF_102
362     // change 39 (WK18): testcase description extension
363     with {extension "Description: 5.2.2 Service Guide
update via broadcast channel (same fragment id, lower version number)"}
364
365     /**
366     *
367     * @desc
368     * <p>Service Guide Update with additional fragments
via broadcast channel</p>
369     * <p>Test Case Description</p>
370     * <p>with a terminal having received a Service
Guide and listening to the broadcast Service Guide Delivery Channel.</p>
371     * <p>when the terminal receives an updated Service
Guide containing an additional content fragment</p>
372     * <p>then the terminal presents the information of
the updated Service Guide</p>
373     * <p>Specification Reference</p>
374     * <p>[BCAST10-ESG] Section 5.4.2.1.1, 5.5 1st
bullet item</p>

```

```

329         if (PX_GET_SG_DISPLAY_MANUALLY) {
330             f_main_ut_GetServiceGuide(PX_SGDD_ID);
331         }
332         f_main_ut_CheckService("TvChannel");
333         f_main_ut_SelectService("TvChannel");
334         f_main_ut_CheckContent("Programm1-version2",
v_DateTime_StartTime, v_DateTime_EndTime);
335
336         // test behavior
337         vc_BCASTTransportID :=
1; // change 07 (WK 6): in order to send
same TransportID
338         v_SGDU :=
{}; // change 10 (WK 6): empty
SGDU to avoid multiple content fragments
339         v_Content.version := v_FragmentVersion;
340         v_Content.Names := {{omit,"Programm1"}};
341         f_addServiceFragment(v_SGDU,
// change 08 (WK 6): SGDU must contain all
v_Service);
fragments
342         f_addContentFragment(v_SGDU, v_Content);
343         f_addScheduleFragment(v_SGDU,
// change 08 (WK 6): SGDU must contain all
v_Schedule);
fragments
344         f_addAccessFragment(v_SGDU,
// change 08 (WK 6): SGDU must contain all
v_Access);
fragments
345         f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp); // change 08 (WK 6): SGDU must contain all fragments
346
347         f_main_ctrl_broadcastServiceGuide(v_SGDU,
omit); // change 11 (WK 6): reset FLUTE; delivers new SGDU
348
349         if (PX_GET_SG_UPDATE_MANUALLY) {
350             f_main_ut_UpdateServiceGuide();
351         }
352         f_main_ut_CheckService("TvChannel");
353         f_main_ut_SelectService("TvChannel");
354         f_main_ut_CheckContent("Programm1-version2",
v_DateTime_StartTime, v_DateTime_EndTime);
355
356         // postamble
357         f_main_ut_PowerOff();
358
359         f_cf_Broadcast_down();
360         f_cf_UpperTester_down();
361     } // end test case TC_BCAST_ESG_CONF_102
362     // change 39 (WK18): testcase description extension
363     with {extension "Description: 5.2.2 Service Guide
update via broadcast channel (same fragment id, lower version number)"}
364
365     /**
366     *
367     * @desc
368     * <p>Service Guide Update with additional fragments
via broadcast channel</p>
369     * <p>Test Case Description</p>
370     * <p>with a terminal having received a Service
Guide and listening to the broadcast Service Guide Delivery Channel.</p>
371     * <p>when the terminal receives an updated Service
Guide containing an additional content fragment</p>
372     * <p>then the terminal presents the information of
the updated Service Guide</p>
373     * <p>Specification Reference</p>
374     * <p>[BCAST10-ESG] Section 5.4.2.1.1, 5.5 1st
bullet item</p>

```



```

375      *      <p>Preconditions</p>
376      *      <p>Service guide cache of terminal is erased</p>
377      *      <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
378      *      <p>Access point information for service guide
entry point is configured in test tool</p>
379      *      <p>This test case uses the following SG fragment
instantiations found in the Appendix:</p>
380      *      <p>Initial service guide:
381      *      <ul>
382      *          <li>Service fragment with
Name="TvChannel"</li>
383      *          <li>Content fragment with
Name="Programme1" , and StartTime and EndTime elements indicating values after
the time of test</li>
384      *          <li>Schedule fragment for content fragment
with same values for startTime and endTime in the presentationWindow element</
li>
385      *          <li>Access fragment for schedule fragment
with BroadcastServiceDelivery element and reference to "Programme1" session
description fragment</li>
386      *          <li>Session description fragment for
"Programme1"</li>
387      *      </ul>
388      *      </p>
389      *      <p>Updated service guide:
390      *      <ul>
391      *          <li>Service fragment with
Name="TvChannel"</li>
392      *          <li>Content fragment with
Name="Programme1" , and StartTime and EndTime elements indicating values after
the time of test</li>
393      *          <li>Schedule fragment for "Programme1"
content fragment with same values for startTime and endTime in the
presentationWindow element</li>
394      *          <li>Access fragment for "Programme1"
schedule fragment with BroadcastServiceDelivery element and a reference to
"Programme1" session description fragment</li>
395      *          <li>Session description fragment for
"Programme1"</li>
396      *          <li>Content fragment with
Name="Programme2" , and different StartTime and EndTime elements indicating
values after the time of test</li>
397      *          <li>Schedule fragment for "Programme2"
content fragment with same different values for startTime and endTime in the
presentationWindow element</li>
398      *          <li>Access fragment for "Programme2"
schedule fragment with BroadcastServiceDelivery element and reference to
"Programme2" session description fragment</li>
399      *          <li>Session description fragment for
"Programme2" </li>
400      *      </ul>
401      *      </p>
402      *      <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
403      *      <p>Test Procedure</p>
404      *      <p>
405      *      <ul>
406      *          <li>Set up the test tool to produce
initial BCAST service guide announcement and delivery using broadcast channel.</
li>
407      *          <li>Activate the BCAST application of the
terminal</li>
408      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
409      *          <li>Browse the SG on the terminal</li>

```

```

375      *      <p>Preconditions</p>
376      *      <p>Service guide cache of terminal is erased</p>
377      *      <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
378      *      <p>Access point information for service guide
entry point is configured in test tool</p>
379      *      <p>This test case uses the following SG fragment
instantiations found in the Appendix:</p>
380      *      <p>Initial service guide:
381      *      <ul>
382      *          <li>Service fragment with
Name="TvChannel"</li>
383      *          <li>Content fragment with
Name="Programme1" , and StartTime and EndTime elements indicating values after
the time of test</li>
384      *          <li>Schedule fragment for content fragment
with same values for startTime and endTime in the presentationWindow element</
li>
385      *          <li>Access fragment for schedule fragment
with BroadcastServiceDelivery element and reference to "Programme1" session
description fragment</li>
386      *          <li>Session description fragment for
"Programme1"</li>
387      *      </ul>
388      *      </p>
389      *      <p>Updated service guide:
390      *      <ul>
391      *          <li>Service fragment with
Name="TvChannel"</li>
392      *          <li>Content fragment with
Name="Programme1" , and StartTime and EndTime elements indicating values after
the time of test</li>
393      *          <li>Schedule fragment for "Programme1"
content fragment with same values for startTime and endTime in the
presentationWindow element</li>
394      *          <li>Access fragment for "Programme1"
schedule fragment with BroadcastServiceDelivery element and a reference to
"Programme1" session description fragment</li>
395      *          <li>Session description fragment for
"Programme1"</li>
396      *          <li>Content fragment with
Name="Programme2" , and different StartTime and EndTime elements indicating
values after the time of test</li>
397      *          <li>Schedule fragment for "Programme2"
content fragment with same different values for startTime and endTime in the
presentationWindow element</li>
398      *          <li>Access fragment for "Programme2"
schedule fragment with BroadcastServiceDelivery element and reference to
"Programme2" session description fragment</li>
399      *          <li>Session description fragment for
"Programme2" </li>
400      *      </ul>
401      *      </p>
402      *      <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
403      *      <p>Test Procedure</p>
404      *      <p>
405      *      <ul>
406      *          <li>Set up the test tool to produce
initial BCAST service guide announcement and delivery using broadcast channel.</
li>
407      *          <li>Activate the BCAST application of the
terminal</li>
408      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
409      *          <li>Browse the SG on the terminal</li>

```

```

410      *          <li>Set up the test tool to produce
updated BCAST service guide announcement and delivery using broadcast channel.</
li>
411      *          <li>Request from BCAST application on
terminal to update the service guide (optional)</li>
412      *          <li>Browse the SG on the terminal</li>
413      *          </ul>
414      *          </p>
415      *          <p>Pass-Criteria</p>
416      *          <p>The following should be visible to the end
user after the initial delivery of the SG
417      *          <ul>
418      *          <li>There is a service "TvChannel"
associated with a program "Programme1" as well as start and end time values
(There is a "TvChannel" that contains "Programme1" scheduled from startTime to
endTime)</li>
419      *          </ul>
420      *          </p>
421      *          <p>The following should be visible to the end
user after the delivery of the updated SG
422      *          <ul>
423      *          <li>There is a service "TvChannel"
associated with programs "Programme1" and "Programme2" as well as different
start and end time values(There is a "TvChannel" that contains "Programme1" and
"Programme2" scheduled from different startTime to endTimes)</li>
424      *          </ul>
425      *          </p>
426      *          @verdict
427      *          pass in case the client pass the conformance test.
428      *          @verdict
429      *          fail in case the client does not pass the
conformance test.
430      *          @verdict
431      *          inconc in case a guard timer expires.
432      */
433      testcase TC_BCAST_ESG_CONF_103() runs on
BCastMainComponent system BCastPlatformComponent {
434          // init components and ports
435          f_createComponents();
436          f_cf_Broadcast_up();
437          f_cf_UpperTester_up();
438
439          // preamble
440          f_startCommonUtPreamble();
441
442          var ServiceGuideDeliveryUnit v_SGDU := {};
443          var UInt32 v_FragmentVersion := 1;
444
445          f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour);
// change 01 (WK 6): global time
definition
446
447          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime);
// change 01 (WK 6): global time
definition
448          var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime);
// change 01 (WK 6): global time definition
449          var DateTime v_DateTime_StartTime2 := "";
450          var DateTime v_DateTime_EndTime2 := "";
451
452          var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
453              PX_SDP_VIDEO_PROG_1_REF_ID,
454              PX_SDP_VIDEO_PROG_1
455          ));
456

```

```

410      *          <li>Set up the test tool to produce
updated BCAST service guide announcement and delivery using broadcast channel.</
li>
411      *          <li>Request from BCAST application on
terminal to update the service guide (optional)</li>
412      *          <li>Browse the SG on the terminal</li>
413      *          </ul>
414      *          </p>
415      *          <p>Pass-Criteria</p>
416      *          <p>The following should be visible to the end
user after the initial delivery of the SG
417      *          <ul>
418      *          <li>There is a service "TvChannel"
associated with a program "Programme1" as well as start and end time values
(There is a "TvChannel" that contains "Programme1" scheduled from startTime to
endTime)</li>
419      *          </ul>
420      *          </p>
421      *          <p>The following should be visible to the end
user after the delivery of the updated SG
422      *          <ul>
423      *          <li>There is a service "TvChannel"
associated with programs "Programme1" and "Programme2" as well as different
start and end time values(There is a "TvChannel" that contains "Programme1" and
"Programme2" scheduled from different startTime to endTimes)</li>
424      *          </ul>
425      *          </p>
426      *          @verdict
427      *          pass in case the client pass the conformance test.
428      *          @verdict
429      *          fail in case the client does not pass the
conformance test.
430      *          @verdict
431      *          inconc in case a guard timer expires.
432      */
433      testcase TC_BCAST_ESG_CONF_103() runs on
BCastMainComponent system BCastPlatformComponent {
434          // init components and ports
435          f_createComponents();
436          f_cf_Broadcast_up();
437          f_cf_UpperTester_up();
438
439          // preamble
440          f_startCommonUtPreamble();
441
442          var ServiceGuideDeliveryUnit v_SGDU := {};
443          var UInt32 v_FragmentVersion := 1;
444
445          f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour);
// change 01 (WK 6): global time
definition
446
447          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime);
// change 01 (WK 6): global time
definition
448          var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime);
// change 01 (WK 6): global time definition
449          var DateTime v_DateTime_StartTime2 := "";
450          var DateTime v_DateTime_EndTime2 := "";
451
452          var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
453              PX_SDP_VIDEO_PROG_1_REF_ID,
454              PX_SDP_VIDEO_PROG_1
455          ));
456

```

```

457      var AccessTypeType v_AccessType := valueof
458      (m_def_AccessType_bc_sdp(
459          f_getAccessType
460          (PX_BROADCAST_NETWORK_BEARER),
461          PX_SDP_VIDEO_PROG_1_REF_ID
462      ));
463      var ServiceType v_Service := valueof
464      (m_def_Service(
465          PX_SGDU_SERVICE_ID,
466          v_FragmentVersion,
467          "TvChannel",
468          c_basicTv_ServiceType
469      ));
470      var ContentType v_Content := valueof
471      (m_def_Content(
472          PX_SGDU_CONTENT_ID_PROG_1,
473          v_FragmentVersion,
474          "Programm1",
475          PX_SGDU_SERVICE_ID,
476          v_DateTime_StartTime,
477          v_DateTime_EndTime
478      ));
479      var ScheduleType v_Schedule := valueof
480      (m_def_Schedule(
481          PX_SGDU_SCHEDULE_ID_PROG_1,
482          v_FragmentVersion,
483          PX_SGDU_SERVICE_ID,
484          PX_SGDU_CONTENT_ID_PROG_1,
485          v_startTime, // change 01 (WK 6):
486          global time definition
487          v_endTime // change 01 (WK 6):
488          global time definition
489      ));
490      var AccessType v_Access := valueof(m_def_Access(
491          PX_SGDU_ACCESS_ID_PROG_1,
492          v_FragmentVersion,
493          v_AccessType,
494          PX_SGDU_SERVICE_ID,
495          PX_SGDU_SCHEDULE_ID_PROG_1,
496          c_class_SG
497      ));
498      f_addServiceFragment(v_SGDU, v_Service);
499      f_addContentFragment(v_SGDU, v_Content);
500      f_addScheduleFragment(v_SGDU, v_Schedule);
501      f_addAccessFragment(v_SGDU, v_Access);
502      f_addSdpFragment(v_SGDU, v_FragmentVersion,
503      v_Sdp);
504      f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
505      f_main_ut_RunBCastApplication();
506      if (PX_GET_SG_DISPLAY_MANUALLY) {
507          f_main_ut_GetServiceGuide(PX_SGDD_ID);
508      }
509      f_main_ut_CheckService("TvChannel");
510      f_main_ut_SelectService("TvChannel");
511      f_main_ut_CheckContent("Programm1",
512      v_DateTime_StartTime, v_DateTime_EndTime);
513      // test behavior

```

```

457      var AccessTypeType v_AccessType := valueof
458      (m_def_AccessType_bc_sdp(
459          f_getAccessType
460          (PX_BROADCAST_NETWORK_BEARER),
461          PX_SDP_VIDEO_PROG_1_REF_ID
462      ));
463      var ServiceType v_Service := valueof
464      (m_def_Service(
465          PX_SGDU_SERVICE_ID,
466          v_FragmentVersion,
467          "TvChannel",
468          c_basicTv_ServiceType
469      ));
470      var ContentType v_Content := valueof
471      (m_def_Content(
472          PX_SGDU_CONTENT_ID_PROG_1,
473          v_FragmentVersion,
474          "Programm1",
475          PX_SGDU_SERVICE_ID,
476          v_DateTime_StartTime,
477          v_DateTime_EndTime
478      ));
479      var ScheduleType v_Schedule := valueof
480      (m_def_Schedule(
481          PX_SGDU_SCHEDULE_ID_PROG_1,
482          v_FragmentVersion,
483          PX_SGDU_SERVICE_ID,
484          PX_SGDU_CONTENT_ID_PROG_1,
485          v_startTime, // change 01 (WK 6):
486          global time definition
487          v_endTime // change 01 (WK 6):
488          global time definition
489      ));
490      var AccessType v_Access := valueof(m_def_Access(
491          PX_SGDU_ACCESS_ID_PROG_1,
492          v_FragmentVersion,
493          v_AccessType,
494          PX_SGDU_SERVICE_ID,
495          PX_SGDU_SCHEDULE_ID_PROG_1,
496          c_class_SG
497      ));
498      f_addServiceFragment(v_SGDU, v_Service);
499      f_addContentFragment(v_SGDU, v_Content);
500      f_addScheduleFragment(v_SGDU, v_Schedule);
501      f_addAccessFragment(v_SGDU, v_Access);
502      f_addSdpFragment(v_SGDU, v_FragmentVersion,
503      v_Sdp);
504      f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
505      f_main_ut_RunBCastApplication();
506      if (PX_GET_SG_DISPLAY_MANUALLY) {
507          f_main_ut_GetServiceGuide(PX_SGDD_ID);
508      }
509      f_main_ut_CheckService("TvChannel");
510      f_main_ut_SelectService("TvChannel");
511      f_main_ut_CheckContent("Programm1",
512      v_DateTime_StartTime, v_DateTime_EndTime);
513      // test behavior

```

```

512 // change 9 (WK18): corrections in start/stop
time for additional program
513 v_startTime := v_endTime;
514 v_endTime := v_endTime + c_one_hour;
515
516 v_DateTime_StartTime2 := f_getDateTime
(v_startTime); // change 01 (WK 6): global
time definition
517 v_DateTime_EndTime2 := f_getDateTime
(v_endTime); // change 01 (WK 6):
global time definition
518
519 v_Sdp := valueof(m_def_SdpFragment(
520     PX_SDP_VIDEO_PROG_2_REF_ID,
521     PX_SDP_VIDEO_PROG_2
522 ));
523
524 v_AccessType := valueof(m_def_AccessType_bc_sdp(
525     f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
526     PX_SDP_VIDEO_PROG_2_REF_ID
527 ));
528
529 v_Content := valueof(m_def_Content(
530     PX_SGDU_CONTENT_ID_PROG_2,
531     v_FragmentVersion,
532     "Programme2",
533     PX_SGDU_SERVICE_ID,
534     v_DateTime_StartTime2,
535     v_DateTime_EndTime2
536 ));
537
538 v_Schedule := valueof(m_def_Schedule(
539     PX_SGDU_SCHEDULE_ID_PROG_2,
540     v_FragmentVersion,
541     PX_SGDU_SERVICE_ID,
542     PX_SGDU_CONTENT_ID_PROG_2,
543     v_startTime, // change 01 (WK 6):
global time definition
544     v_endTime // change 01 (WK 6):
global time definition
545 ));
546
547 v_Access := valueof(m_def_Access(
548     PX_SGDU_ACCESS_ID_PROG_2,
549     v_FragmentVersion,
550     v_AccessType,
551     PX_SGDU_SERVICE_ID,
552     PX_SGDU_SCHEDULE_ID_PROG_2,
553     c_class_SG
554 ));
555
556 // v_SGDU :=
{}; // change 12 (WK 6): SGDU must
contain additional fragments for 'Programme2'
557
558 f_addContentFragment(v_SGDU, v_Content);
559 f_addScheduleFragment(v_SGDU, v_Schedule);
560 f_addAccessFragment(v_SGDU, v_Access);
561 f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp);
562
563 f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
564

```

```

512 // change 9 (WK18): corrections in start/stop
time for additional program
513 v_startTime := v_endTime;
514 v_endTime := v_endTime + c_one_hour;
515
516 v_DateTime_StartTime2 := f_getDateTime
(v_startTime); // change 01 (WK 6): global
time definition
517 v_DateTime_EndTime2 := f_getDateTime
(v_endTime); // change 01 (WK 6):
global time definition
518
519 v_Sdp := valueof(m_def_SdpFragment(
520     PX_SDP_VIDEO_PROG_2_REF_ID,
521     PX_SDP_VIDEO_PROG_2
522 ));
523
524 v_AccessType := valueof(m_def_AccessType_bc_sdp(
525     f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
526     PX_SDP_VIDEO_PROG_2_REF_ID
527 ));
528
529 v_Content := valueof(m_def_Content(
530     PX_SGDU_CONTENT_ID_PROG_2,
531     v_FragmentVersion,
532     "Programme2",
533     PX_SGDU_SERVICE_ID,
534     v_DateTime_StartTime2,
535     v_DateTime_EndTime2
536 ));
537
538 v_Schedule := valueof(m_def_Schedule(
539     PX_SGDU_SCHEDULE_ID_PROG_2,
540     v_FragmentVersion,
541     PX_SGDU_SERVICE_ID,
542     PX_SGDU_CONTENT_ID_PROG_2,
543     v_startTime, // change 01 (WK 6):
global time definition
544     v_endTime // change 01 (WK 6):
global time definition
545 ));
546
547 v_Access := valueof(m_def_Access(
548     PX_SGDU_ACCESS_ID_PROG_2,
549     v_FragmentVersion,
550     v_AccessType,
551     PX_SGDU_SERVICE_ID,
552     PX_SGDU_SCHEDULE_ID_PROG_2,
553     c_class_SG
554 ));
555
556 // v_SGDU :=
{}; // change 12 (WK 6): SGDU must
contain additional fragments for 'Programme2'
557
558 f_addContentFragment(v_SGDU, v_Content);
559 f_addScheduleFragment(v_SGDU, v_Schedule);
560 f_addAccessFragment(v_SGDU, v_Access);
561 f_addSdpFragment(v_SGDU, v_FragmentVersion,
v_Sdp);
562
563 f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
564

```

```

565         if (PX_GET_SG_UPDATE_MANUALLY) {
566             f_main_ut_UpdateServiceGuide();
567         }
568         f_main_ut_CheckService("TvChannel");
569         f_main_ut_SelectService("TvChannel");
570         f_main_ut_CheckContent("Programme1",
v_DateTime_StartTime, v_DateTime_EndTime);
571         f_main_ut_CheckContent("Programme2",
v_DateTime_StartTime2, v_DateTime_EndTime2);
572
573         // postamble
574         f_main_ut_PowerOff();
575
576         f_cf_Broadcast_down();
577         f_cf_UpperTester_down();
578     } // end test case TC_BCAST_ESG_CONF_103
579     // change 39 (WK18): testcase description extension
580     with {extension "Description: 5.2.3 Service Guide
Update with additional fragments via broadcast channel"}
581
582     /**
583     *
584     * @desc
585     * <p>GZIP compression of Service Guide Delivery
Unit on broadcast channel</p>
586     * <p>Test Case Description</p>
587     * <p>with a terminal listening to the broadcast
Service Guide Delivery Channel.</p>
588     * <p>when the terminal receives a GZIP compressed
Service Guide</p>
589     * <p>then the terminal presents the information of
the Service Guide</p>
590     * <p>Specification Reference</p>
591     * <p>[BCAST10-ESG] Section 5.4.1.4.</p>
592     * <p>Preconditions</p>
593     * <p>Service guide cache of terminal is erased</p>
594     * <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
595     * <p>Access point information for service guide
entry point is configured in test tool</p>
596     * <p>This test cases uses the following SG fragment
instantiations found in the Appendix:
597     * <ul>
598     * <li>Service fragment with
Name="TvChannel"</li>
599     * <li>Content fragment with
Name="Programme1"</li>
600     * <li>Schedule fragment for content
fragment</li>
601     * <li>Access fragment for schedule fragment</li>
602     * </ul>
603     * <p>
604     * <p>All fragments are packaged in one SGDU, which
is GZIP compressed.</p>
605     * <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
606     * <p>Test Procedure</p>
607     * <p>
608     * <ul>
609     * <li>Set up the test tool to produce BCAST
service guide announcement and delivery using broadcast channel.</li>
610     * <li>Activate the BCAST application of the
terminal</li>
611     * <li>Request from BCAST application on
terminal to get the service guide (optional)</li>

```

```

565         if (PX_GET_SG_UPDATE_MANUALLY) {
566             f_main_ut_UpdateServiceGuide();
567         }
568         f_main_ut_CheckService("TvChannel");
569         f_main_ut_SelectService("TvChannel");
570         f_main_ut_CheckContent("Programme1",
v_DateTime_StartTime, v_DateTime_EndTime);
571         f_main_ut_CheckContent("Programme2",
v_DateTime_StartTime2, v_DateTime_EndTime2);
572
573         // postamble
574         f_main_ut_PowerOff();
575
576         f_cf_Broadcast_down();
577         f_cf_UpperTester_down();
578     } // end test case TC_BCAST_ESG_CONF_103
579     // change 39 (WK18): testcase description extension
580     with {extension "Description: 5.2.3 Service Guide
Update with additional fragments via broadcast channel"}
581
582     /**
583     *
584     * @desc
585     * <p>GZIP compression of Service Guide Delivery
Unit on broadcast channel</p>
586     * <p>Test Case Description</p>
587     * <p>with a terminal listening to the broadcast
Service Guide Delivery Channel.</p>
588     * <p>when the terminal receives a GZIP compressed
Service Guide</p>
589     * <p>then the terminal presents the information of
the Service Guide</p>
590     * <p>Specification Reference</p>
591     * <p>[BCAST10-ESG] Section 5.4.1.4.</p>
592     * <p>Preconditions</p>
593     * <p>Service guide cache of terminal is erased</p>
594     * <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
595     * <p>Access point information for service guide
entry point is configured in test tool</p>
596     * <p>This test cases uses the following SG fragment
instantiations found in the Appendix:
597     * <ul>
598     * <li>Service fragment with
Name="TvChannel"</li>
599     * <li>Content fragment with
Name="Programme1"</li>
600     * <li>Schedule fragment for content
fragment</li>
601     * <li>Access fragment for schedule fragment</li>
602     * </ul>
603     * <p>
604     * <p>All fragments are packaged in one SGDU, which
is GZIP compressed.</p>
605     * <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
606     * <p>Test Procedure</p>
607     * <p>
608     * <ul>
609     * <li>Set up the test tool to produce BCAST
service guide announcement and delivery using broadcast channel.</li>
610     * <li>Activate the BCAST application of the
terminal</li>
611     * <li>Request from BCAST application on
terminal to get the service guide (optional)</li>

```

```

612      *      <li>Browse the SG on the terminal</li>
613      *      </ul>
614      *      </p>
615      *      <p>Pass-Criteria</p>
616      *      <p></p>
617      *      <p></p>
618      *      <p>The following should be visible to the end user
619      *      <ul>
620      *      <li>There is a service "TvChannel"
associated with programme "Programmel"(There is a "TvChannel" that contains
"Programmel")</li>
621      *      </ul>
622      *      </p>
623      *      @verdict
624      *      pass in case the client pass the conformance test.
625      *      @verdict
626      *      fail in case the client does not pass the
conformance test.
627      *      @verdict
628      *      inconc in case a guard timer expires.
629      */
630      testcase TC_BCAST_ESG_CONF_104() runs on
BCastMainComponent system BCastPlatformComponent {
631          // init components and ports
632          f_createComponents();
633          f_cf_Broadcast_up();
634          f_cf_UpperTester_up();
635
636          f_startCommonUtPreamble();
637
638          // test behavior
639          var ServiceGuideDeliveryUnit v_SGDU := {};
640          var UInt32 v_Version := 1;
641
642          f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour);          // change 01 (WK 6): global time
definition
643
644          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime);          // change 01 (WK 6): global
time definition
645
646          var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime);          // change 01 (WK 6): global time
definition
647
648          var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
649              PX_SDP_VIDEO_PROG_1_REF_ID,
650              PX_SDP_VIDEO_PROG_1
651          ));
652          var AccessTypeType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
653              f_getAccessType
654              (PX_BROADCAST_NETWORK_BEARER),
655              PX_SDP_VIDEO_PROG_1_REF_ID
656          ));
657          var ServiceType v_Service := valueof
(m_def_Service(
658              PX_SGDU_SERVICE_ID,
659              v_Version,
660              "TvChannel",
661              c_basicTv_ServiceType
662          ));
663

```

```

612      *      <li>Browse the SG on the terminal</li>
613      *      </ul>
614      *      </p>
615      *      <p>Pass-Criteria</p>
616      *      <p></p>
617      *      <p></p>
618      *      <p>The following should be visible to the end user
619      *      <ul>
620      *      <li>There is a service "TvChannel"
associated with programme "Programmel"(There is a "TvChannel" that contains
"Programmel")</li>
621      *      </ul>
622      *      </p>
623      *      @verdict
624      *      pass in case the client pass the conformance test.
625      *      @verdict
626      *      fail in case the client does not pass the
conformance test.
627      *      @verdict
628      *      inconc in case a guard timer expires.
629      */
630      testcase TC_BCAST_ESG_CONF_104() runs on
BCastMainComponent system BCastPlatformComponent {
631          // init components and ports
632          f_createComponents();
633          f_cf_Broadcast_up();
634          f_cf_UpperTester_up();
635
636          f_startCommonUtPreamble();
637
638          // test behavior
639          var ServiceGuideDeliveryUnit v_SGDU := {};
640          var UInt32 v_Version := 1;
641
642          f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour);          // change 01 (WK 6): global time
definition
643
644          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime);          // change 01 (WK 6): global
time definition
645
646          var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime);          // change 01 (WK 6): global time
definition
647
648          var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
649              PX_SDP_VIDEO_PROG_1_REF_ID,
650              PX_SDP_VIDEO_PROG_1
651          ));
652          var AccessTypeType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
653              f_getAccessType
654              (PX_BROADCAST_NETWORK_BEARER),
655              PX_SDP_VIDEO_PROG_1_REF_ID
656          ));
657          var ServiceType v_Service := valueof
(m_def_Service(
658              PX_SGDU_SERVICE_ID,
659              v_Version,
660              "TvChannel",
661              c_basicTv_ServiceType
662          ));
663

```

```

664      var ContentType v_Content := valueof
665      (m_def_Content(
666          PX_SGDU_CONTENT_ID_PROG_1,
667          v_Version,
668          "Programm1",
669          PX_SGDU_SERVICE_ID,
670          v_DateTime_StartTime,
671          v_DateTime_EndTime
672      ));
673      var ScheduleType v_Schedule := valueof
674      (m_def_Schedule(
675          PX_SGDU_SCHEDULE_ID_PROG_1,
676          v_Version,
677          PX_SGDU_SERVICE_ID,
678          PX_SGDU_CONTENT_ID_PROG_1,
679          v_startTime, // change 01 (WK 6):
680          global time definition
681          v_endTime // change 01 (WK 6):
682          global time definition
683      ));
684      var AccessType v_Access := valueof(m_def_Access(
685          PX_SGDU_ACCESS_ID_PROG_1,
686          v_Version,
687          v_AccessType,
688          PX_SGDU_SERVICE_ID,
689          PX_SGDU_SCHEDULE_ID_PROG_1,
690          c_class_SG
691      ));
692      f_addServiceFragment(v_SGDU, v_Service);
693      f_addContentFragment(v_SGDU, v_Content);
694      f_addScheduleFragment(v_SGDU, v_Schedule);
695      f_addAccessFragment(v_SGDU, v_Access);
696      f_addSdpFragment(v_SGDU, v_Version, v_Sdp);
697      f_main_ctrl_broadcastServiceGuide(v_SGDU,
698      "gzip");
699      f_main_ut_RunBCastApplication();
700      if (PX_GET_SG_DISPLAY_MANUALLY) {
701          f_main_ut_GetServiceGuide(PX_SGDD_ID);
702      }
703      f_main_ut_CheckService("TvChannel");
704      f_main_ut_SelectService("TvChannel");
705      f_main_ut_CheckContent("Programm1",
706      v_DateTime_StartTime, v_DateTime_EndTime );
707      // postamble
708      f_main_ut_PowerOff();
709      f_cf_Broadcast_down();
710      f_cf_UpperTester_down();
711      }// end test case TC_BCAST_ESG_CONF_104
712      // change 39 (WK18): testcase description extension
713      with {extension "Description: 5.2.4 GZIP compression of
714      Service Guide Delivery Unit on broadcast channel"}
715      /**
716      *
717      * @desc
718      * <p>Service provisioning with Content hierarchy</p>
719      * <p>Test Case Description</p>

```

```

664      var ContentType v_Content := valueof
665      (m_def_Content(
666          PX_SGDU_CONTENT_ID_PROG_1,
667          v_Version,
668          "Programm1",
669          PX_SGDU_SERVICE_ID,
670          v_DateTime_StartTime,
671          v_DateTime_EndTime
672      ));
673      var ScheduleType v_Schedule := valueof
674      (m_def_Schedule(
675          PX_SGDU_SCHEDULE_ID_PROG_1,
676          v_Version,
677          PX_SGDU_SERVICE_ID,
678          PX_SGDU_CONTENT_ID_PROG_1,
679          v_startTime, // change 01 (WK 6):
680          global time definition
681          v_endTime // change 01 (WK 6):
682          global time definition
683      ));
684      var AccessType v_Access := valueof(m_def_Access(
685          PX_SGDU_ACCESS_ID_PROG_1,
686          v_Version,
687          v_AccessType,
688          PX_SGDU_SERVICE_ID,
689          PX_SGDU_SCHEDULE_ID_PROG_1,
690          c_class_SG
691      ));
692      f_addServiceFragment(v_SGDU, v_Service);
693      f_addContentFragment(v_SGDU, v_Content);
694      f_addScheduleFragment(v_SGDU, v_Schedule);
695      f_addAccessFragment(v_SGDU, v_Access);
696      f_addSdpFragment(v_SGDU, v_Version, v_Sdp);
697      f_main_ctrl_broadcastServiceGuide(v_SGDU,
698      "gzip");
699      f_main_ut_RunBCastApplication();
700      if (PX_GET_SG_DISPLAY_MANUALLY) {
701          f_main_ut_GetServiceGuide(PX_SGDD_ID);
702      }
703      f_main_ut_CheckService("TvChannel");
704      f_main_ut_SelectService("TvChannel");
705      f_main_ut_CheckContent("Programm1",
706      v_DateTime_StartTime, v_DateTime_EndTime );
707      // postamble
708      f_main_ut_PowerOff();
709      f_cf_Broadcast_down();
710      f_cf_UpperTester_down();
711      }// end test case TC_BCAST_ESG_CONF_104
712      // change 39 (WK18): testcase description extension
713      with {extension "Description: 5.2.4 GZIP compression of
714      Service Guide Delivery Unit on broadcast channel"}
715      /**
716      *
717      * @desc
718      * <p>Service provisioning with Content hierarchy</p>
719      * <p>Test Case Description</p>

```

721 * <p>with a terminal listening to the Service Guide
Delivery Channel and for a data stream on the broadcast channel</p>
722 * <p>when the terminal receives an Service Guide
containing service fragments with two content fragments that follow each other
in time</p>
723 * <p>then the terminal presents the information of
the Service Guide and is able to receive the data stream correctly</p>
724 * <p>Specification Reference</p>
725 * <p>[BCAST10-ESG] Section 5.1.2.1, 5.1.2.2,
5.1.2.3, 5.1.2.4, 5.1.2.5.1, 5.4.2.1, 6.1.1</p>
726 * <p>Preconditions</p>
727 * <p>Service guide cache of terminal is erased.</p>
728 * <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
729 * <p>Access point information for service guide
entry point is configured in test tool</p>
730 * <p>This test case uses the following SG fragment
instantiations found in the Appendix:
731 *
732 * Service fragment with
Name="TvChannel"
733 * Content fragment for service fragment
with Name="Programme1" with StartTime and EndTime elements indicating values
within test execution time that result in a play time of three minutes
734 * Schedule fragment for content fragment
"Programme1" with same values for startTime and endTime in the
presentationWindow element
735 * Access fragment for "Programme1"
schedule fragment with BroadcastServiceDelivery element and a reference to
"Programme1" session description fragment
736 * Session description fragment for
"Programme1"
737 * Content fragment for service fragment
with Name="Programme2" with StartTime and EndTime elements indicating values
within test execution time but after the ones of "Programme1"
738 * Schedule fragment for content fragment
"Programme2" with same values for startTime and endTime in the
presentationWindow element
739 * Access fragment for "Programme2"
schedule fragment with BroadcastServiceDelivery element and a reference to
"Programme2" session description fragment
740 * Session description fragment for
"Programme2"
741 *
742 * </p>
743 * <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
744 * <p>Test Procedure</p>
745 * <p>
746 *
747 * Set up the test tool to produce
initial BCAST service guide announcement and delivery using broadcast channel.
748 * Setup the test tool to stream audio
and video for "Programme1" and "Programme2" via the broadcast channel
749 * Activate the BCAST application of the
terminal
750 * Request from BCAST application on
terminal to get the service guide (optional)
751 * Browse the SG on the terminal
752 * Access "TvChannel" on terminal
753 *
754 * </p>
755 * <p>The used video codec is:
756 *
757 * H.264 (AVC)

721 * <p>with a terminal listening to the Service Guide
Delivery Channel and for a data stream on the broadcast channel</p>
722 * <p>when the terminal receives an Service Guide
containing service fragments with two content fragments that follow each other
in time</p>
723 * <p>then the terminal presents the information of
the Service Guide and is able to receive the data stream correctly</p>
724 * <p>Specification Reference</p>
725 * <p>[BCAST10-ESG] Section 5.1.2.1, 5.1.2.2,
5.1.2.3, 5.1.2.4, 5.1.2.5.1, 5.4.2.1, 6.1.1</p>
726 * <p>Preconditions</p>
727 * <p>Service guide cache of terminal is erased.</p>
728 * <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
729 * <p>Access point information for service guide
entry point is configured in test tool</p>
730 * <p>This test case uses the following SG fragment
instantiations found in the Appendix:
731 *
732 * Service fragment with
Name="TvChannel"
733 * Content fragment for service fragment
with Name="Programme1" with StartTime and EndTime elements indicating values
within test execution time that result in a play time of three minutes
734 * Schedule fragment for content fragment
"Programme1" with same values for startTime and endTime in the
presentationWindow element
735 * Access fragment for "Programme1"
schedule fragment with BroadcastServiceDelivery element and a reference to
"Programme1" session description fragment
736 * Session description fragment for
"Programme1"
737 * Content fragment for service fragment
with Name="Programme2" with StartTime and EndTime elements indicating values
within test execution time but after the ones of "Programme1"
738 * Schedule fragment for content fragment
"Programme2" with same values for startTime and endTime in the
presentationWindow element
739 * Access fragment for "Programme2"
schedule fragment with BroadcastServiceDelivery element and a reference to
"Programme2" session description fragment
740 * Session description fragment for
"Programme2"
741 *
742 * </p>
743 * <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
744 * <p>Test Procedure</p>
745 * <p>
746 *
747 * Set up the test tool to produce
initial BCAST service guide announcement and delivery using broadcast channel.
748 * Setup the test tool to stream audio
and video for "Programme1" and "Programme2" via the broadcast channel
749 * Activate the BCAST application of the
terminal
750 * Request from BCAST application on
terminal to get the service guide (optional)
751 * Browse the SG on the terminal
752 * Access "TvChannel" on terminal
753 *
754 * </p>
755 * <p>The used video codec is:
756 *
757 * H.264 (AVC)


```

758      *      </ul>
759      *      </p>
760      *      <p>Pass-Criteria</p>
761      *      <p>The following should be visible to the end
user after the delivery of the SG
762      *      <ul>
763      *      <li>There is a service "TvChannel"
associated with a program "Programmel" followed three minutes later by a
program "Programme2"(There is a "TvChannel" that contains "Programmel"
scheduled three minutes prior to "Programme2")</li>
764      *      <li>Terminal shows "Programmel" for three
minutes which is followed by "Programme2"</li>
765      *      </ul>
766      *      </p>
767      *      @verdict
768      *      pass in case the client pass the conformance test.
769      *      @verdict
770      *      fail in case the client does not pass the
conformance test.
771      *      @verdict
772      *      inconc in case a guard timer expires.
773      */
774      testcase TC_BCAST_ESG_CONF_105() runs on
BCastMainComponent system BCastPlatformComponent {
775          // init components and ports
776          f_createComponents();
777          f_cf_Broadcast_up();
778          f_cf_Data_up();
779          f_cf_UpperTester_up();
780
781          // preamble
782
783          // change 14 (WK18): dynamic setting of
destination IP
784          f_main_data_startStreamingFile(0,
PX_FILE_VIDEO_PROG1, PX_SDP_VIDEO_PROG_1, PX_DATA_STREAMING_DESTINATION_IP_1);
785          //f_main_data_startStreamingFile(1,
PX_FILE_AUDIO_PROG_1, PX_SDP_AUDIO_PROG_1); // hint: 1st programm broadcasts
video and audio
786
787          // change 14 (WK18): dynamic setting of
destination IP
788          f_main_data_startStreamingFile(1,
PX_FILE_VIDEO_PROG2, PX_SDP_VIDEO_PROG_2, PX_DATA_STREAMING_DESTINATION_IP_2);
789          //f_main_data_startStreamingFile(3,
PX_FILE_AUDIO_PROG_2, PX_SDP_AUDIO_PROG_2); // hint: 2st programm broadcasts
video and audio
790
791          f_startCommonUtPreamble();
792
793          // test behavior
794          // current time
795          var ServiceGuideDeliveryUnit v_SGDU := {};
796          var UInt32 v_Version := 1;
797
798          f_main_ctrl_InitStartEndTime(0,c_one_hour +
PX_PLAY_TIME); // change 9 (WK18): corrections in start/stop time for
additional program          // change 01 (WK 6): global
time definition
799
800          // change 9 (WK18): corrections in start/stop
time for additional program
801          // time after v_endTime
802          var UInt v_NTP_StartTime2 := v_startTime +
PX_PLAY_TIME;          // change 01 (WK 6):
global time definition

```

```

758      *      </ul>
759      *      </p>
760      *      <p>Pass-Criteria</p>
761      *      <p>The following should be visible to the end
user after the delivery of the SG
762      *      <ul>
763      *      <li>There is a service "TvChannel"
associated with a program "Programmel" followed three minutes later by a
program "Programme2"(There is a "TvChannel" that contains "Programmel"
scheduled three minutes prior to "Programme2")</li>
764      *      <li>Terminal shows "Programmel" for three
minutes which is followed by "Programme2"</li>
765      *      </ul>
766      *      </p>
767      *      @verdict
768      *      pass in case the client pass the conformance test.
769      *      @verdict
770      *      fail in case the client does not pass the
conformance test.
771      *      @verdict
772      *      inconc in case a guard timer expires.
773      */
774      testcase TC_BCAST_ESG_CONF_105() runs on
BCastMainComponent system BCastPlatformComponent {
775          // init components and ports
776          f_createComponents();
777          f_cf_Broadcast_up();
778          f_cf_Data_up();
779          f_cf_UpperTester_up();
780
781          // preamble
782
783          // change 14 (WK18): dynamic setting of
destination IP
784          f_main_data_startStreamingFile(0,
PX_FILE_VIDEO_PROG1, PX_SDP_VIDEO_PROG_1, PX_DATA_STREAMING_DESTINATION_IP_1);
785          //f_main_data_startStreamingFile(1,
PX_FILE_AUDIO_PROG_1, PX_SDP_AUDIO_PROG_1); // hint: 1st programm broadcasts
video and audio
786
787          // change 14 (WK18): dynamic setting of
destination IP
788          f_main_data_startStreamingFile(1,
PX_FILE_VIDEO_PROG2, PX_SDP_VIDEO_PROG_2, PX_DATA_STREAMING_DESTINATION_IP_2);
789          //f_main_data_startStreamingFile(3,
PX_FILE_AUDIO_PROG_2, PX_SDP_AUDIO_PROG_2); // hint: 2st programm broadcasts
video and audio
790
791          f_startCommonUtPreamble();
792
793          // test behavior
794          // current time
795          var ServiceGuideDeliveryUnit v_SGDU := {};
796          var UInt32 v_Version := 1;
797
798          f_main_ctrl_InitStartEndTime(0,c_one_hour +
PX_PLAY_TIME); // change 9 (WK18): corrections in start/stop time for
additional program          // change 01 (WK 6): global
time definition
799
800          // change 9 (WK18): corrections in start/stop
time for additional program
801          // time after v_endTime
802          var UInt v_NTP_StartTime2 := v_startTime +
PX_PLAY_TIME;          // change 01 (WK 6):
global time definition

```

```

803 // 1 hour duration;
804 var UInt v_NTP_EndTime2 :=
v_endTime; // change 01 (WK 6):
global time definition
805
806 var DateTime v_DateTime_StartTime1 :=
f_getDateTime(v_startTime); // change 01 (WK 6): global time
definition
807
808 var DateTime v_DateTime_EndTime1 :=
f_getDateTime(v_NTP_StartTime2); // change 9 (WK18): corrections in start/stop
time for additional program // change 01 (WK 6): global time
definition
808
809 var DateTime v_DateTime_StartTime2 :=
f_getDateTime(v_NTP_StartTime2);
810 var DateTime v_DateTime_EndTime2 :=
f_getDateTime(v_NTP_EndTime2);
811
812 var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
813 PX_SDP_VIDEO_PROG_1_REF_ID,
814 PX_SDP_VIDEO_PROG_1
815 ));
816
817 var AccessType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
818 f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
819 PX_SDP_VIDEO_PROG_1_REF_ID
820 ));
821
822 var ServiceType v_Service := valueof
(m_def_Service(
823 PX_SGDU_SERVICE_ID,
824 v_Version,
825 "TvChannel",
826 c_basicTv_ServiceType
827 ));
828
829 var ContentType v_Content := valueof
(m_def_Content(
830 PX_SGDU_CONTENT_ID_PROG_1,
831 v_Version,
832 "Programm1",
833 PX_SGDU_SERVICE_ID,
834 v_DateTime_StartTime1,
835 v_DateTime_EndTime1
836 ));
837
838 var ScheduleType v_Schedule := valueof
(m_def_Schedule(
839 PX_SGDU_SCHEDULE_ID_PROG_1,
840 v_Version,
841 PX_SGDU_SERVICE_ID,
842 PX_SGDU_CONTENT_ID_PROG_1,
843
v_startTime, // change 01 (WK 6):
global time definition
844
v_NTP_StartTime2 // change 9 (WK18):
corrections in start/stop time for additional program // change 01 (WK 6):
global time definition
845
));
846
847 var AccessType v_Access := valueof(m_def_Access(
848 PX_SGDU_ACCESS_ID_PROG_1,
849 v_Version,

```

```

803 // 1 hour duration;
804 var UInt v_NTP_EndTime2 :=
v_endTime; // change 01 (WK 6):
global time definition
805
806 var DateTime v_DateTime_StartTime1 :=
f_getDateTime(v_startTime); // change 01 (WK 6): global time
definition
807
808 var DateTime v_DateTime_EndTime1 :=
f_getDateTime(v_NTP_StartTime2); // change 9 (WK18): corrections in start/stop
time for additional program // change 01 (WK 6): global time
definition
808
809 var DateTime v_DateTime_StartTime2 :=
f_getDateTime(v_NTP_StartTime2);
810 var DateTime v_DateTime_EndTime2 :=
f_getDateTime(v_NTP_EndTime2);
811
812 var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
813 PX_SDP_VIDEO_PROG_1_REF_ID,
814 PX_SDP_VIDEO_PROG_1
815 ));
816
817 var AccessType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
818 f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
819 PX_SDP_VIDEO_PROG_1_REF_ID
820 ));
821
822 var ServiceType v_Service := valueof
(m_def_Service(
823 PX_SGDU_SERVICE_ID,
824 v_Version,
825 "TvChannel",
826 c_basicTv_ServiceType
827 ));
828
829 var ContentType v_Content := valueof
(m_def_Content(
830 PX_SGDU_CONTENT_ID_PROG_1,
831 v_Version,
832 "Programm1",
833 PX_SGDU_SERVICE_ID,
834 v_DateTime_StartTime1,
835 v_DateTime_EndTime1
836 ));
837
838 var ScheduleType v_Schedule := valueof
(m_def_Schedule(
839 PX_SGDU_SCHEDULE_ID_PROG_1,
840 v_Version,
841 PX_SGDU_SERVICE_ID,
842 PX_SGDU_CONTENT_ID_PROG_1,
843
v_startTime, // change 01 (WK 6):
global time definition
844
v_NTP_StartTime2 // change 9 (WK18):
corrections in start/stop time for additional program // change 01 (WK 6):
global time definition
845
));
846
847 var AccessType v_Access := valueof(m_def_Access(
848 PX_SGDU_ACCESS_ID_PROG_1,
849 v_Version,

```

```

850         v_AccessType,
851         PX_SGDU_SERVICE_ID,
852         PX_SGDU_SCHEDULE_ID_PROG_1,
853         c_class_ST // change 35 (WK18): Check
Service Class of Access Fragments: ST, SG, FC: ST when access references video,
FC for file distribution, otherwise SG.
854     ));
855
856     var SdpFragment v_Sdp2 := valueof
(m_def_SdpFragment(
857         PX_SDP_VIDEO_PROG_2_REF_ID,
858         PX_SDP_VIDEO_PROG_2
859     ));
860
861     var AccessTypeType v_AccessType2 := valueof
(m_def_AccessType_bc_sdp(
862         f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
863         PX_SDP_VIDEO_PROG_2_REF_ID
864     ));
865
866     var ContentType v_Content2 := valueof
(m_def_Content(
867         PX_SGDU_CONTENT_ID_PROG_2,
868         v_Version,
869         "Programme2",
870         PX_SGDU_SERVICE_ID,
871         v_DateTime_StartTime2,
872         v_DateTime_EndTime2
873     ));
874
875     var ScheduleType v_Schedule2 := valueof
(m_def_Schedule(
876         PX_SGDU_SCHEDULE_ID_PROG_2,
877         v_Version,
878         PX_SGDU_SERVICE_ID,
879         PX_SGDU_CONTENT_ID_PROG_2,
880         v_NTP_StartTime2,
881         v_NTP_EndTime2
882     ));
883
884     var AccessType v_Access2 := valueof(m_def_Access
(
885         PX_SGDU_ACCESS_ID_PROG_2,
886         v_Version,
887         v_AccessType2, // // change 10 (WK18):
content hierarchy -> video 2 needs another SDP description
888         PX_SGDU_SERVICE_ID,
889         PX_SGDU_SCHEDULE_ID_PROG_2,
890         c_class_ST // change 35 (WK18): Check
Service Class of Access Fragments: ST, SG, FC: ST when access references video,
FC for file distribution, otherwise SG.
891     ));
892
893     f_addServiceFragment(v_SGDU, v_Service);
894     f_addContentFragment(v_SGDU, v_Content);
895     f_addScheduleFragment(v_SGDU, v_Schedule);
896     f_addAccessFragment(v_SGDU, v_Access);
897     f_addContentFragment(v_SGDU, v_Content2);
898     f_addScheduleFragment(v_SGDU, v_Schedule2);
899     f_addAccessFragment(v_SGDU, v_Access2);
900     f_addSdpFragment(v_SGDU, v_Version, v_Sdp);
901     f_addSdpFragment(v_SGDU, v_Version, v_Sdp2);
902
903     f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
904

```

```

850         v_AccessType,
851         PX_SGDU_SERVICE_ID,
852         PX_SGDU_SCHEDULE_ID_PROG_1,
853         c_class_ST // change 35 (WK18): Check
Service Class of Access Fragments: ST, SG, FC: ST when access references video,
FC for file distribution, otherwise SG.
854     ));
855
856     var SdpFragment v_Sdp2 := valueof
(m_def_SdpFragment(
857         PX_SDP_VIDEO_PROG_2_REF_ID,
858         PX_SDP_VIDEO_PROG_2
859     ));
860
861     var AccessTypeType v_AccessType2 := valueof
(m_def_AccessType_bc_sdp(
862         f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
863         PX_SDP_VIDEO_PROG_2_REF_ID
864     ));
865
866     var ContentType v_Content2 := valueof
(m_def_Content(
867         PX_SGDU_CONTENT_ID_PROG_2,
868         v_Version,
869         "Programme2",
870         PX_SGDU_SERVICE_ID,
871         v_DateTime_StartTime2,
872         v_DateTime_EndTime2
873     ));
874
875     var ScheduleType v_Schedule2 := valueof
(m_def_Schedule(
876         PX_SGDU_SCHEDULE_ID_PROG_2,
877         v_Version,
878         PX_SGDU_SERVICE_ID,
879         PX_SGDU_CONTENT_ID_PROG_2,
880         v_NTP_StartTime2,
881         v_NTP_EndTime2
882     ));
883
884     var AccessType v_Access2 := valueof(m_def_Access
(
885         PX_SGDU_ACCESS_ID_PROG_2,
886         v_Version,
887         v_AccessType2, // // change 10 (WK18):
content hierarchy -> video 2 needs another SDP description
888         PX_SGDU_SERVICE_ID,
889         PX_SGDU_SCHEDULE_ID_PROG_2,
890         c_class_ST // change 35 (WK18): Check
Service Class of Access Fragments: ST, SG, FC: ST when access references video,
FC for file distribution, otherwise SG.
891     ));
892
893     f_addServiceFragment(v_SGDU, v_Service);
894     f_addContentFragment(v_SGDU, v_Content);
895     f_addScheduleFragment(v_SGDU, v_Schedule);
896     f_addAccessFragment(v_SGDU, v_Access);
897     f_addContentFragment(v_SGDU, v_Content2);
898     f_addScheduleFragment(v_SGDU, v_Schedule2);
899     f_addAccessFragment(v_SGDU, v_Access2);
900     f_addSdpFragment(v_SGDU, v_Version, v_Sdp);
901     f_addSdpFragment(v_SGDU, v_Version, v_Sdp2);
902
903     f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
904

```

```

905      f_main_ut_RunBCastApplication();
906      if (PX_GET_SG_DISPLAY_MANUALLY) {
907          f_main_ut_GetServiceGuide(PX_SGDD_ID);
908      }
909      f_main_ut_CheckService("TvChannel");
910      f_main_ut_SelectService("TvChannel");
911      f_main_ut_CheckContent("Programme1",
v_DateTime_StartTime1, v_DateTime_EndTime1);
912      f_main_ut_CheckContent("Programme2",
v_DateTime_StartTime2, v_DateTime_EndTime2);
913      f_main_ut_UseService("TvChannel");
914
915      f_main_ut_CheckVideo("Programme1");
916      // change 9 (WK18): corrections in start/stop
time for additional programm
917      var UInt v_NTP_NOW := f_getNTPTime();
918      var integer v_TimeToWait := v_NTP_StartTime2 -
v_NTP_NOW;
919
920      if (v_TimeToWait > 0) {
921          // waiting end of play time of
Programme1
922          timer t_wait;
923          t_wait.start(int2float(v_TimeToWait));
924          t_wait.timeout;
925      }
926
927      f_main_ut_CheckVideo("Programme2");
928
929      // postamble
930      f_main_ut_PowerOff();
931
932      f_main_data_stopStreaming(0);
933      //f_main_data_stopStreaming(1);
934      f_main_data_stopStreaming(1);
935      //f_main_data_stopStreaming(3);
936
937      f_cf_Broadcast_down();
938      f_cf_Data_down();
939      f_cf_UpperTester_down();
940      }// end test case TC_BCAST_ESG_CONF_105
941      // change 39 (WK18): testcase description extension
942      with {extension "Description: 5.2.5 Service
provisioning with Content hierarchy"}
943
944      /**
945       *
946       * @desc
947       * <p>Service guide with Preview Data via broadcast
channel</p>
948       * <p>Test Case Description</p>
949       * <p>with a terminal listening to the broadcast
Service Guide Delivery Channel.</p>
950       * <p>when the terminal receives a Service Guide
containing a service and preview data fragments indicating a picture reference
and alternative text</p>
951       * <p>then the terminal presents the service and the
picture or the alternative text</p>
952       * <p>Specification Reference</p>
953       * <p>[BCAST10-ESG] Section 5.1.2.9</p>
954       * <p>Preconditions</p>
955       * <p>Service guide cache of terminal is erased</p>
956       * <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
957       * <p>Access point information for service guide
entry point is configured in test tool</p>

```

```

905      f_main_ut_RunBCastApplication();
906      if (PX_GET_SG_DISPLAY_MANUALLY) {
907          f_main_ut_GetServiceGuide(PX_SGDD_ID);
908      }
909      f_main_ut_CheckService("TvChannel");
910      f_main_ut_SelectService("TvChannel");
911      f_main_ut_CheckContent("Programme1",
v_DateTime_StartTime1, v_DateTime_EndTime1);
912      f_main_ut_CheckContent("Programme2",
v_DateTime_StartTime2, v_DateTime_EndTime2);
913      f_main_ut_UseService("TvChannel");
914
915      f_main_ut_CheckVideo("Programme1");
916      // change 9 (WK18): corrections in start/stop
time for additional programm
917      var UInt v_NTP_NOW := f_getNTPTime();
918      var integer v_TimeToWait := v_NTP_StartTime2 -
v_NTP_NOW;
919
920      if (v_TimeToWait > 0) {
921          // waiting end of play time of
Programme1
922          timer t_wait;
923          t_wait.start(int2float(v_TimeToWait));
924          t_wait.timeout;
925      }
926
927      f_main_ut_CheckVideo("Programme2");
928
929      // postamble
930      f_main_ut_PowerOff();
931
932      f_main_data_stopStreaming(0);
933      //f_main_data_stopStreaming(1);
934      f_main_data_stopStreaming(1);
935      //f_main_data_stopStreaming(3);
936
937      f_cf_Broadcast_down();
938      f_cf_Data_down();
939      f_cf_UpperTester_down();
940      }// end test case TC_BCAST_ESG_CONF_105
941      // change 39 (WK18): testcase description extension
942      with {extension "Description: 5.2.5 Service
provisioning with Content hierarchy"}
943
944      /**
945       *
946       * @desc
947       * <p>Service guide with Preview Data via broadcast
channel</p>
948       * <p>Test Case Description</p>
949       * <p>with a terminal listening to the broadcast
Service Guide Delivery Channel.</p>
950       * <p>when the terminal receives a Service Guide
containing a service and preview data fragments indicating a picture reference
and alternative text</p>
951       * <p>then the terminal presents the service and the
picture or the alternative text</p>
952       * <p>Specification Reference</p>
953       * <p>[BCAST10-ESG] Section 5.1.2.9</p>
954       * <p>Preconditions</p>
955       * <p>Service guide cache of terminal is erased</p>
956       * <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
957       * <p>Access point information for service guide
entry point is configured in test tool</p>

```

```

958      *          <p>Preview icon is delivered using a FLUTE
file delivery session.</p>
959      *          <p>This test cases uses the following SG fragment
instantiations and files found in the Appendix:
960      *          <ul>
961      *          <li>Service fragment with
Name="TvChannel"</li>
962      *          <li>PreviewData fragment for service
fragment with PictureUri and AlternativeText element set to "TV Channel icon"</
li>
963      *          <li>Access fragment for previewData
fragment with reference to session description fragment</li>
964      *          <li>Session description fragment with
parameters for broadcast file delivery session</li>
965      *          <li>Image file "TvChannelIcon.jpg"</li>
966      *          </ul>
967      *          </p>
968      *          <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
969      *          <p>Set the FLUTE session to contain an FDT-
instance with 'File'-element, where 'Content-Location= "TvChannelIcon.jpg"'.</p>
970      *          <p>Test Procedure</p>
971      *          <p>
972      *          <ul>
973      *          <li>Set up the test tool to produce BCAST
service guide announcement and delivery using broadcast channel.</li>
974      *          <li>Setup the test tool to deliver the
image file "TvChannelIcon.jpg" using a file delivery session as described by
SDP.</li>
975      *          <li>Activate the BCAST application of the
terminal</li>
976      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
977      *          <li>Browse the SG on the terminal</li>
978      *          </ul>
979      *          </p>
980      *          <p>Pass-Criteria</p>
981      *          <p>
982      *          <ul>
983      *          <li>Terminal associates and displays
either the icon "TvChannelIcon.jpg".or alternative text "TV Channel icon" with
the service "TvChannel"</li>
984      *          </ul>
985      *          </p>
986      *          @verdict
987      *          pass in case the client pass the conformance test.
988      *          @verdict
989      *          fail in case the client does not pass the
conformance test.
990      *          @verdict
991      *          inconc in case a guard timer expires.
992      */
993      testcase TC_BCAST_ESG_CONF_106() runs on
BCastMainComponent system BCastPlatformComponent {
994      *          // init components and ports
995      *          f_createComponents();
996      *          f_cf_Broadcast_up();
997      *          f_cf_Data_up();
998      *          f_cf_UpperTester_up();
999      *          f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour);
1000      *          // change 01 (WK 6): global time
definition
1001      *          // preamble
1002      *          f_startCommonUtPreamble();
1003

```

```

958      *          <p>Preview icon is delivered using a FLUTE
file delivery session.</p>
959      *          <p>This test cases uses the following SG fragment
instantiations and files found in the Appendix:
960      *          <ul>
961      *          <li>Service fragment with
Name="TvChannel"</li>
962      *          <li>PreviewData fragment for service
fragment with PictureUri and AlternativeText element set to "TV Channel icon"</
li>
963      *          <li>Access fragment for previewData
fragment with reference to session description fragment</li>
964      *          <li>Session description fragment with
parameters for broadcast file delivery session</li>
965      *          <li>Image file "TvChannelIcon.jpg"</li>
966      *          </ul>
967      *          </p>
968      *          <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
969      *          <p>Set the FLUTE session to contain an FDT-
instance with 'File'-element, where 'Content-Location= "TvChannelIcon.jpg"'.</p>
970      *          <p>Test Procedure</p>
971      *          <p>
972      *          <ul>
973      *          <li>Set up the test tool to produce BCAST
service guide announcement and delivery using broadcast channel.</li>
974      *          <li>Setup the test tool to deliver the
image file "TvChannelIcon.jpg" using a file delivery session as described by
SDP.</li>
975      *          <li>Activate the BCAST application of the
terminal</li>
976      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
977      *          <li>Browse the SG on the terminal</li>
978      *          </ul>
979      *          </p>
980      *          <p>Pass-Criteria</p>
981      *          <p>
982      *          <ul>
983      *          <li>Terminal associates and displays
either the icon "TvChannelIcon.jpg".or alternative text "TV Channel icon" with
the service "TvChannel"</li>
984      *          </ul>
985      *          </p>
986      *          @verdict
987      *          pass in case the client pass the conformance test.
988      *          @verdict
989      *          fail in case the client does not pass the
conformance test.
990      *          @verdict
991      *          inconc in case a guard timer expires.
992      */
993      testcase TC_BCAST_ESG_CONF_106() runs on
BCastMainComponent system BCastPlatformComponent {
994      *          // init components and ports
995      *          f_createComponents();
996      *          f_cf_Broadcast_up();
997      *          f_cf_Data_up();
998      *          f_cf_UpperTester_up();
999      *          f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour);
1000      *          // change 01 (WK 6): global time
definition
1001      *          // preamble
1002      *          f_startCommonUtPreamble();
1003

```

```
1004 // test behavior
1005 f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour); // change 01 (WK 6): global time
definition
1006
1007 var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime); // change 01 (WK 6): global time
definition
1008 var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime); // change 01 (WK 6): global time definition
1009
1010 // change 2 (WK18): file distribution as well
as preview data
1011 var TransferFileList v_FileList := {
1012 {
1013 Content_Location :=
PX_DATA_SESSION, // change 34 (WK18): rename "TVChannelIcon" SDP session into
"Data_Session", rename respective SDP PIXIT paramter as well
1014 Content_Type := "image/jpeg",
1015 Content-Encoding := omit,
1016 file := omit // use the content-
Location as file reference
1017 }
1018 };
1019 var UInt v_TransferId := 4;
1020 // change 34 (WK18): rename "TVChannelIcon" SDP
session into "Data_Session", rename respective SDP PIXIT paramter as well
1021 f_main_data_startFileTransfer(v_TransferId,
PX_SDP_DATA_SESSION, v_FileList);
1022
1023 var ServiceGuideDeliveryUnit v_SGDU := {};
1024 var UInt32 v_Version := 1;
1025
1026 // change 2 (WK18): file distribution as well
as preview data
1027 // change 34 (WK18): rename "TVChannelIcon" SDP
session into "Data_Session", rename respective SDP PIXIT paramter as well
1028 var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
1029 PX_SDP_DATA_SESSION_REF_ID,
1030 PX_SDP_DATA_SESSION
1031 ));
1032
1033 // change 34 (WK18): rename "TVChannelIcon" SDP
session into "Data_Session", rename respective SDP PIXIT paramter as well
1034 var AccessTypeType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
1035 f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
1036 PX_SDP_DATA_SESSION_REF_ID // change 2
(WK18): file distribution as well as preview data
1037 ));
1038
1039 var ServiceType v_Service := valueof
(m_def_Service(
1040 PX_SGDU_SERVICE_ID,
1041 v_Version,
1042 "TvChannel",
1043 c_basicTv_ServiceType
1044 ));
1045 // change 2 (WK18): file distribution as well
as preview data
1046 v_Service.PreviewDataReferences := {{idRef :=
PX_SGDU_PREVIEW_DATA_ID, usage := 2}};
1047
```

```
1004 // test behavior
1005 f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour); // change 01 (WK 6): global time
definition
1006
1007 var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime); // change 01 (WK 6): global time
definition
1008 var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime); // change 01 (WK 6): global time definition
1009
1010 // change 2 (WK18): file distribution as well
as preview data
1011 var TransferFileList v_FileList := {
1012 {
1013 Content_Location :=
PX_DATA_SESSION, // change 34 (WK18): rename "TVChannelIcon" SDP session into
"Data_Session", rename respective SDP PIXIT paramter as well
1014 Content_Type := "image/jpeg",
1015 Content-Encoding := omit,
1016 file := omit // use the content-
Location as file reference
1017 }
1018 };
1019 var UInt v_TransferId := 4;
1020 // change 34 (WK18): rename "TVChannelIcon" SDP
session into "Data_Session", rename respective SDP PIXIT paramter as well
1021 f_main_data_startFileTransfer(v_TransferId,
PX_SDP_DATA_SESSION, v_FileList);
1022
1023 var ServiceGuideDeliveryUnit v_SGDU := {};
1024 var UInt32 v_Version := 1;
1025
1026 // change 2 (WK18): file distribution as well
as preview data
1027 // change 34 (WK18): rename "TVChannelIcon" SDP
session into "Data_Session", rename respective SDP PIXIT paramter as well
1028 var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
1029 PX_SDP_DATA_SESSION_REF_ID,
1030 PX_SDP_DATA_SESSION
1031 ));
1032
1033 // change 34 (WK18): rename "TVChannelIcon" SDP
session into "Data_Session", rename respective SDP PIXIT paramter as well
1034 var AccessTypeType v_AccessType := valueof
(m_def_AccessType_bc_sdp(
1035 f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
1036 PX_SDP_DATA_SESSION_REF_ID // change 2
(WK18): file distribution as well as preview data
1037 ));
1038
1039 var ServiceType v_Service := valueof
(m_def_Service(
1040 PX_SGDU_SERVICE_ID,
1041 v_Version,
1042 "TvChannel",
1043 c_basicTv_ServiceType
1044 ));
1045 // change 2 (WK18): file distribution as well
as preview data
1046 v_Service.PreviewDataReferences := {{idRef :=
PX_SGDU_PREVIEW_DATA_ID, usage := 2}};
1047
```

```

1048      var PreviewDataType v_PreviewData := valueof
1049      (m_def_picture_PreviewData(
1050          PX_SGDU_PREVIEW_DATA_ID,
1051          v_Version,
1052          PX_DATA_SESSION, // change 34 (WK18):
1053          rename "TVChannelIcon" SDP session into "Data_Session", rename respective SDP
1054          PIXIT paramter as well
1055          {omit,"TV Channel icon"},
1056          "image/jpeg" // change 2 (WK18): file
1057          distribution as well as preview data
1058          ));
1059      var AccessType v_Access := valueof
1060      (m_def_Access_without_Ref(
1061          PX_SGDU_ACCESS_ID_PROG_1,
1062          v_Version,
1063          v_AccessType,
1064          PX_SGDU_SERVICE_ID,
1065          c_class_FC // change 35 (WK18): Check
1066          Service Class of Access Fragments: ST, SG, FC: ST when access references video,
1067          FC for file distribution, otherwise SG.
1068          ));
1069      f_addServiceFragment(v_SGDU, v_Service);
1070      f_addPreviewDataFragment(v_SGDU, v_PreviewData);
1071      f_addAccessFragment(v_SGDU, v_Access);
1072      f_addSdpFragment(v_SGDU, v_Version, v_Sdp);
1073      f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
1074      f_main_ut_RunBCastApplication();
1075      if (PX_GET_SG_DISPLAY_MANUALLY) {
1076          f_main_ut_GetServiceGuide(PX_SGDD_ID);
1077      }
1078      f_main_ut_CheckService("TvChannel");
1079      f_main_ut_CheckPreview("icon TvChannelIcon.jpg
1080      or text "TV Channel icon");
1081      // postamble
1082      f_main_ut_PowerOff();
1083      f_main_data_stopFileTransfer(v_TransferId);
1084      f_cf_Broadcast_down();
1085      f_cf_Data_down();
1086      f_cf_UpperTester_down();
1087      }// end test case TC_BCAST_ESG_CONF_106
1088      // change 39 (WK18): testcase description extension
1089      with {extension "Description: 5.2.6 Service guide with
1090      Preview Data via broadcast channel"}
1091      /**
1092      *
1093      * @desc
1094      * <p>Service Guide with dual audio streams via
1095      broadcast channel</p>
1096      * <p>Test Case Description</p>
1097      * <p>with a terminal listening to the broadcast
1098      Service Guide Delivery Channel.</p>
1099      * <p>when the terminal receives Service Guide
1100      containing one content fragment associated with two access fragments pointing
1101      to different audio streams</p>
1102      * <p>then the terminal presents the information of
1103      the Service Guide and is capable of associating the content with both access
1104      fragments.</p>
1105      * <p>Specification Reference</p>

```

```

1048      var PreviewDataType v_PreviewData := valueof
1049      (m_def_picture_PreviewData(
1050          PX_SGDU_PREVIEW_DATA_ID,
1051          v_Version,
1052          PX_DATA_SESSION, // change 34 (WK18):
1053          rename "TVChannelIcon" SDP session into "Data_Session", rename respective SDP
1054          PIXIT paramter as well
1055          {omit,"TV Channel icon"},
1056          "image/jpeg" // change 2 (WK18): file
1057          distribution as well as preview data
1058          ));
1059      var AccessType v_Access := valueof
1060      (m_def_Access_without_Ref(
1061          PX_SGDU_ACCESS_ID_PROG_1,
1062          v_Version,
1063          v_AccessType,
1064          PX_SGDU_SERVICE_ID,
1065          c_class_FC // change 35 (WK18): Check
1066          Service Class of Access Fragments: ST, SG, FC: ST when access references video,
1067          FC for file distribution, otherwise SG.
1068          ));
1069      f_addServiceFragment(v_SGDU, v_Service);
1070      f_addPreviewDataFragment(v_SGDU, v_PreviewData);
1071      f_addAccessFragment(v_SGDU, v_Access);
1072      f_addSdpFragment(v_SGDU, v_Version, v_Sdp);
1073      f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
1074      f_main_ut_RunBCastApplication();
1075      if (PX_GET_SG_DISPLAY_MANUALLY) {
1076          f_main_ut_GetServiceGuide(PX_SGDD_ID);
1077      }
1078      f_main_ut_CheckService("TvChannel");
1079      f_main_ut_CheckPreview("icon TvChannelIcon.jpg
1080      or text "TV Channel icon");
1081      // postamble
1082      f_main_ut_PowerOff();
1083      f_main_data_stopFileTransfer(v_TransferId);
1084      f_cf_Broadcast_down();
1085      f_cf_Data_down();
1086      f_cf_UpperTester_down();
1087      }// end test case TC_BCAST_ESG_CONF_106
1088      // change 39 (WK18): testcase description extension
1089      with {extension "Description: 5.2.6 Service guide with
1090      Preview Data via broadcast channel"}
1091      /**
1092      *
1093      * @desc
1094      * <p>Service Guide with dual audio streams via
1095      broadcast channel</p>
1096      * <p>Test Case Description</p>
1097      * <p>with a terminal listening to the broadcast
1098      Service Guide Delivery Channel.</p>
1099      * <p>when the terminal receives Service Guide
1100      containing one content fragment associated with two access fragments pointing
1101      to different audio streams</p>
1102      * <p>then the terminal presents the information of
1103      the Service Guide and is capable of associating the content with both access
1104      fragments.</p>
1105      * <p>Specification Reference</p>

```

```
1098      *      <p>[BCAST10-ESG] Section 5.5.1, 6.1.1, 7.2.1,
Appendix C.3 (informative)</p>
1099      *      <p>Preconditions</p>
1100      *      <p>Service guide cache of terminal is erased</p>
1101      *      <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
1102      *      <p>Access point information for service guide
entry point is configured in test tool</p>
1103      *      <p>This test cases uses the following SG fragment
instantiations and files found in the Appendix:
1104      *      <ul>
1105      *          <li>Service fragment with
Name="TvChannel"</li>
1106      *          <li>Content fragment with
Name="Programme"</li>
1107      *          <li>Schedule fragment for content
fragment</li>
1108      *          <li>Access fragment for schedule fragment
with BroadcastServiceDelivery element and a reference to the "Programme_eng"
session description fragment </li>
1109      *          <li>Session Description fragment
containing SDP for "Programme_eng"(audio language is English)</li>
1110      *          <li>Second schedule fragment for content
fragment</li>
1111      *          <li>Access fragment for second schedule
fragment with BroadcastServiceDelivery element and a reference to the
"Programme_ger" session description fragment</li>
1112      *          <li>Session Description fragment
containing SDP for "Programme_ger"(audio language is German)</li>
1113      *          <li>Terminal provides means for the user
in selecting the audio language.</li>
1114      *      </ul>
1115      *      <p>
1116      *      <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery</p>
1117      *      <p>Test Procedure</p>
1118      *      <p>
1119      *      <ul>
1120      *          <li>Set up the test tool to produce BCAST
service guide announcement and delivery using broadcast channel.</li>
1121      *          <li>Activate the BCAST application of the
terminal</li>
1122      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
1123      *          <li>Browse the SG in the terminal</li>
1124      *          <li>Setup the test tool to stream video,
English audio, and German audio via the broadcast channel as defined by the SDP
fragments</li>
1125      *          <li>Select English as the audio language
for "Programme"</li>
1126      *          <li>Start viewing "Programme"</li>
1127      *          <li>Select German as the audio language
for "Programme"</li>
1128      *          <li>Continue viewing "Programme" </li>
1129      *      </ul>
1130      *      <p>
1131      *      <p>Pass-Criteria</p>
1132      *      <p>The following should be visible and audible to
the end user after the delivery of the SG
1133      *      <ul>
1134      *          <li>There is a service "TvChannel"
associated with a programme "Programme".</li>
1135      *          <li>"Programme" can be viewed by the end
user with English as the audio language.</li>
1136      *          <li>"Programme" can be viewed by the end
user with German as the audio language.</li>
```

```
1098      *      <p>[BCAST10-ESG] Section 5.5.1, 6.1.1, 7.2.1,
Appendix C.3 (informative)</p>
1099      *      <p>Preconditions</p>
1100      *      <p>Service guide cache of terminal is erased</p>
1101      *      <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
1102      *      <p>Access point information for service guide
entry point is configured in test tool</p>
1103      *      <p>This test cases uses the following SG fragment
instantiations and files found in the Appendix:
1104      *      <ul>
1105      *          <li>Service fragment with
Name="TvChannel"</li>
1106      *          <li>Content fragment with
Name="Programme"</li>
1107      *          <li>Schedule fragment for content
fragment</li>
1108      *          <li>Access fragment for schedule fragment
with BroadcastServiceDelivery element and a reference to the "Programme_eng"
session description fragment </li>
1109      *          <li>Session Description fragment
containing SDP for "Programme_eng"(audio language is English)</li>
1110      *          <li>Second schedule fragment for content
fragment</li>
1111      *          <li>Access fragment for second schedule
fragment with BroadcastServiceDelivery element and a reference to the
"Programme_ger" session description fragment</li>
1112      *          <li>Session Description fragment
containing SDP for "Programme_ger"(audio language is German)</li>
1113      *          <li>Terminal provides means for the user
in selecting the audio language.</li>
1114      *      </ul>
1115      *      <p>
1116      *      <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery</p>
1117      *      <p>Test Procedure</p>
1118      *      <p>
1119      *      <ul>
1120      *          <li>Set up the test tool to produce BCAST
service guide announcement and delivery using broadcast channel.</li>
1121      *          <li>Activate the BCAST application of the
terminal</li>
1122      *          <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
1123      *          <li>Browse the SG in the terminal</li>
1124      *          <li>Setup the test tool to stream video,
English audio, and German audio via the broadcast channel as defined by the SDP
fragments</li>
1125      *          <li>Select English as the audio language
for "Programme"</li>
1126      *          <li>Start viewing "Programme"</li>
1127      *          <li>Select German as the audio language
for "Programme"</li>
1128      *          <li>Continue viewing "Programme" </li>
1129      *      </ul>
1130      *      <p>
1131      *      <p>Pass-Criteria</p>
1132      *      <p>The following should be visible and audible to
the end user after the delivery of the SG
1133      *      <ul>
1134      *          <li>There is a service "TvChannel"
associated with a programme "Programme".</li>
1135      *          <li>"Programme" can be viewed by the end
user with English as the audio language.</li>
1136      *          <li>"Programme" can be viewed by the end
user with German as the audio language.</li>
```



```
1137 *      </ul>
1138 *      </p>
1139 * @verdict
1140 *      pass in case the client pass the conformance test.
1141 * @verdict
1142 *      fail in case the client does not pass the
conformance test.
1143 * @verdict
1144 *      inconc in case a guard timer expires.
1145 */
1146 testcase TC_BCAST_ESG_CONF_107() runs on
BCastMainComponent system BCastPlatformComponent {
1147     // init components and ports
1148     f_createComponents();
1149     f_cf_Broadcast_up();
1150     f_cf_UpperTester_up();
1151
1152     // preamble
1153     // change 14 (WK18): dynamic setting of
destination IP
1154     f_main_data_startStreamingFile(0,
PX_FILE_VIDEO_PROG1, PX_SDP_VIDEO_PROG_1, PX_DATA_STREAMING_DESTINATION_IP_1);
1155     f_main_data_startStreamingFile(1,
PX_FILE_AUDIO_ENG, PX_SDP_AUDIO_ENG, PX_DATA_STREAMING_DESTINATION_IP_2);
1156     f_main_data_startStreamingFile(2,
PX_FILE_AUDIO_GER, PX_SDP_AUDIO_GER, PX_DATA_STREAMING_DESTINATION_IP_2);
1157
1158     f_startCommonUtPreamble();
1159
1160     // test behavior
1161     var ServiceGuideDeliveryUnit v_SGDU := {};
1162     var UInt32 v_Version := 1;
1163
1164     f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour); // change 01 (WK 6): global time
definition
1165
1166     var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime); // change 01 (WK 6): global
time definition
1167
1168     var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime); // change 01 (WK 6): global time definition
1169
1170     var SdpFragment v_Sdp1 := valueof
(m_def_SdpFragment(
1171         PX_SDP_VIDEO_PROG_1_REF_ID,
1172         PX_SDP_AUDIO_ENG
1173     ));
1174
1175     var SdpFragment v_Sdp2 := valueof
(m_def_SdpFragment(
1176         PX_SDP_VIDEO_PROG_2_REF_ID,
1177         PX_SDP_AUDIO_GER
1178     ));
1179
1180     var AccessTypeType v_AccessType1 := valueof
(m_def_AccessType_bc_sdp(
1181         f_getAccessType
1182         (PX_BROADCAST_NETWORK_BEARER),
1183         PX_SDP_VIDEO_PROG_1_REF_ID
1184     ));
1185
1186     var AccessTypeType v_AccessType2 := valueof
(m_def_AccessType_bc_sdp(
1187         f_getAccessType
1188         (PX_BROADCAST_NETWORK_BEARER),
```

```
1137 *      </ul>
1138 *      </p>
1139 * @verdict
1140 *      pass in case the client pass the conformance test.
1141 * @verdict
1142 *      fail in case the client does not pass the
conformance test.
1143 * @verdict
1144 *      inconc in case a guard timer expires.
1145 */
1146 testcase TC_BCAST_ESG_CONF_107() runs on
BCastMainComponent system BCastPlatformComponent {
1147     // init components and ports
1148     f_createComponents();
1149     f_cf_Broadcast_up();
1150     f_cf_UpperTester_up();
1151
1152     // preamble
1153     // change 14 (WK18): dynamic setting of
destination IP
1154     f_main_data_startStreamingFile(0,
PX_FILE_VIDEO_PROG1, PX_SDP_VIDEO_PROG_1, PX_DATA_STREAMING_DESTINATION_IP_1);
1155     f_main_data_startStreamingFile(1,
PX_FILE_AUDIO_ENG, PX_SDP_AUDIO_ENG, PX_DATA_STREAMING_DESTINATION_IP_2);
1156     f_main_data_startStreamingFile(2,
PX_FILE_AUDIO_GER, PX_SDP_AUDIO_GER, PX_DATA_STREAMING_DESTINATION_IP_2);
1157
1158     f_startCommonUtPreamble();
1159
1160     // test behavior
1161     var ServiceGuideDeliveryUnit v_SGDU := {};
1162     var UInt32 v_Version := 1;
1163
1164     f_main_ctrl_InitStartEndTime(c_one_hour,
c_one_hour); // change 01 (WK 6): global time
definition
1165
1166     var DateTime v_DateTime_StartTime :=
f_getDateTime(v_startTime); // change 01 (WK 6): global
time definition
1167
1168     var DateTime v_DateTime_EndTime := f_getDateTime
(v_endTime); // change 01 (WK 6): global time definition
1169
1170     var SdpFragment v_Sdp1 := valueof
(m_def_SdpFragment(
1171         PX_SDP_VIDEO_PROG_1_REF_ID,
1172         PX_SDP_AUDIO_ENG
1173     ));
1174
1175     var SdpFragment v_Sdp2 := valueof
(m_def_SdpFragment(
1176         PX_SDP_VIDEO_PROG_2_REF_ID,
1177         PX_SDP_AUDIO_GER
1178     ));
1179
1180     var AccessTypeType v_AccessType1 := valueof
(m_def_AccessType_bc_sdp(
1181         f_getAccessType
1182         (PX_BROADCAST_NETWORK_BEARER),
1183         PX_SDP_VIDEO_PROG_1_REF_ID
1184     ));
1185
1186     var AccessTypeType v_AccessType2 := valueof
(m_def_AccessType_bc_sdp(
1187         f_getAccessType
1188         (PX_BROADCAST_NETWORK_BEARER),
```

```

1186          PX_SDP_VIDEO_PROG_2_REF_ID
1187      ));
1188
1189      var ServiceType v_Service := valueof
1190      (m_def_Service(
1191          PX_SGDU_SERVICE_ID,
1192          v_Version,
1193          "TvChannel",
1194          c_basicTv_ServiceType
1195      ));
1196
1197      var ContentType v_Content := valueof
1198      (m_def_Content(
1199          PX_SGDU_CONTENT_ID_PROG_1,
1200          v_Version,
1201          "Programme",
1202          PX_SGDU_SERVICE_ID,
1203          v_DateTime_StartTime,
1204          v_DateTime_EndTime
1205      ));
1206
1207      var ScheduleType v_Schedule1 := valueof
1208      (m_def_Schedule(
1209          PX_SGDU_SCHEDULE_ID_PROG_1,
1210          v_Version,
1211          PX_SGDU_SERVICE_ID,
1212          PX_SGDU_CONTENT_ID_PROG_1,
1213          v_startTime, // change
1214          01 (WK 6): global time definition
1215          v_endTime // change
1216          01 (WK 6): global time definition
1217      ));
1218
1219      var ScheduleType v_Schedule2 := valueof
1220      (m_def_Schedule(
1221          PX_SGDU_SCHEDULE_ID_PROG_2,
1222          v_Version,
1223          PX_SGDU_SERVICE_ID,
1224          PX_SGDU_CONTENT_ID_PROG_2,
1225          v_startTime, // change
1226          01 (WK 6): global time definition
1227          v_endTime // change
1228          01 (WK 6): global time definition
1229      ));
1230
1231      var AccessType v_Access1 := valueof(m_def_Access
1232      (
1233          PX_SGDU_ACCESS_ID_PROG_1,
1234          v_Version,
1235          v_AccessType1,
1236          PX_SGDU_SERVICE_ID,
1237          PX_SGDU_SCHEDULE_ID_PROG_1,
1238          c_class_ST // change 35 (WK18): Check
1239          Service Class of Access Fragments: ST, SG, FC: ST when access references video,
1240          FC for file distribution, otherwise SG.
1241      ));
1242
1243      var AccessType v_Access2 := valueof(m_def_Access
1244      (
1245          PX_SGDU_ACCESS_ID_PROG_2,
1246          v_Version,

```

```

1186          PX_SDP_VIDEO_PROG_2_REF_ID
1187      ));
1188
1189      var ServiceType v_Service := valueof
1190      (m_def_Service(
1191          PX_SGDU_SERVICE_ID,
1192          v_Version,
1193          "TvChannel",
1194          c_basicTv_ServiceType
1195      ));
1196
1197      var ContentType v_Content := valueof
1198      (m_def_Content(
1199          PX_SGDU_CONTENT_ID_PROG_1,
1200          v_Version,
1201          "Programme",
1202          PX_SGDU_SERVICE_ID,
1203          v_DateTime_StartTime,
1204          v_DateTime_EndTime
1205      ));
1206
1207      var ScheduleType v_Schedule1 := valueof
1208      (m_def_Schedule(
1209          PX_SGDU_SCHEDULE_ID_PROG_1,
1210          v_Version,
1211          PX_SGDU_SERVICE_ID,
1212          PX_SGDU_CONTENT_ID_PROG_1,
1213          v_startTime, // change
1214          01 (WK 6): global time definition
1215          v_endTime // change
1216          01 (WK 6): global time definition
1217      ));
1218
1219      var ScheduleType v_Schedule2 := valueof
1220      (m_def_Schedule(
1221          PX_SGDU_SCHEDULE_ID_PROG_2,
1222          v_Version,
1223          PX_SGDU_SERVICE_ID,
1224          PX_SGDU_CONTENT_ID_PROG_2,
1225          v_startTime, // change
1226          01 (WK 6): global time definition
1227          v_endTime // change
1228          01 (WK 6): global time definition
1229      ));
1230
1231      var AccessType v_Access1 := valueof(m_def_Access
1232      (
1233          PX_SGDU_ACCESS_ID_PROG_1,
1234          v_Version,
1235          v_AccessType1,
1236          PX_SGDU_SERVICE_ID,
1237          PX_SGDU_SCHEDULE_ID_PROG_1,
1238          c_class_ST // change 35 (WK18): Check
1239          Service Class of Access Fragments: ST, SG, FC: ST when access references video,
1240          FC for file distribution, otherwise SG.
1241      ));
1242
1243      var AccessType v_Access2 := valueof(m_def_Access
1244      (
1245          PX_SGDU_ACCESS_ID_PROG_2,
1246          v_Version,

```

```
1236         v_AccessType2,
1237         PX_SGDU_SERVICE_ID,
1238         PX_SGDU_SCHEDULE_ID_PROG_2,
1239         c_class_ST // change 35 (WK18): Check
Service Class of Access Fragments: ST, SG, FC: ST when access references video,
FC for file distribution, otherwise SG.
1240     ));
1241
1242     f_addServiceFragment(v_SGDU, v_Service);
1243     f_addContentFragment(v_SGDU, v_Content);
1244     f_addScheduleFragment(v_SGDU, v_Schedule1);
1245     f_addAccessFragment(v_SGDU, v_Access1);
1246     f_addScheduleFragment(v_SGDU, v_Schedule2);
1247     f_addAccessFragment(v_SGDU, v_Access2);
1248     f_addSdpFragment(v_SGDU, v_Version, v_Sdp1);
1249     f_addSdpFragment(v_SGDU, v_Version, v_Sdp2);
1250
1251     f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
1252
1253     f_main_ut_RunBCastApplication();
1254     if (PX_GET_SG_DISPLAY_MANUALLY) {
1255         f_main_ut_GetServiceGuide(PX_SGDD_ID);
1256     }
1257     f_main_ut_CheckService("TvChannel");
1258     f_main_ut_SelectService("TvChannel");
1259     f_main_ut_CheckContent("Programme",
1260 v_DateTime_StartTime, v_DateTime_EndTime);
1261
1262     f_main_ut_SelectLanguage(e_audio, "English");
1263     f_main_ut_UseService("TvChannel");
1264     f_main_ut_CheckLanguage("Programme", e_audio,
"English");
1265
1266     f_main_ut_SelectLanguage(e_audio, "German");
1267     f_main_ut_UseService("TvChannel");
1268     f_main_ut_CheckLanguage("Programme", e_audio,
"German");
1269
1270     // postamble
1271     f_main_ut_PowerOff();
1272     f_main_data_stopStreaming(0);
1273     f_main_data_stopStreaming(1);
1274     f_main_data_stopStreaming(2);
1275
1276     f_cf_Broadcast_down();
1277     f_cf_UpperTester_down();
1278 } // end test case TC_BCAST_ESG_CONF_107
1279 // change 39 (WK18): testcase description extension
with {extension "Description: 5.2.7 Service Guide with
dual audio streams via broadcast channel"}
1280
1281 /**
1282  *
1283  * @desc
1284  * <p>Service guide with purchase information</p>
1285  * <p>Test Case Description</p>
1286  * <p>with a terminal listening to the broadcast
Service Guide Delivery Channel.</p>
1287  * <p>when the terminal receives Service Guide
containing purchase item, purchase channel and purchase data fragments</p>
1288  * <p>then the terminal presents the information of
the Service Guide</p>
1289  * <p>Specification Reference</p>
1290  * <p>[BCAST10-ESG] Section 5.1.2.6, 5.1.2.7,
5.1.2.8, 5.5.1, 6.1.1<br>[BCAST10-sServices]</p>
1291  * <p>Preconditions</p>
```

```
1236         v_AccessType2,
1237         PX_SGDU_SERVICE_ID,
1238         PX_SGDU_SCHEDULE_ID_PROG_2,
1239         c_class_ST // change 35 (WK18): Check
Service Class of Access Fragments: ST, SG, FC: ST when access references video,
FC for file distribution, otherwise SG.
1240     ));
1241
1242     f_addServiceFragment(v_SGDU, v_Service);
1243     f_addContentFragment(v_SGDU, v_Content);
1244     f_addScheduleFragment(v_SGDU, v_Schedule1);
1245     f_addAccessFragment(v_SGDU, v_Access1);
1246     f_addScheduleFragment(v_SGDU, v_Schedule2);
1247     f_addAccessFragment(v_SGDU, v_Access2);
1248     f_addSdpFragment(v_SGDU, v_Version, v_Sdp1);
1249     f_addSdpFragment(v_SGDU, v_Version, v_Sdp2);
1250
1251     f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
1252
1253     f_main_ut_RunBCastApplication();
1254     if (PX_GET_SG_DISPLAY_MANUALLY) {
1255         f_main_ut_GetServiceGuide(PX_SGDD_ID);
1256     }
1257     f_main_ut_CheckService("TvChannel");
1258     f_main_ut_SelectService("TvChannel");
1259     f_main_ut_CheckContent("Programme",
1260 v_DateTime_StartTime, v_DateTime_EndTime);
1261
1262     f_main_ut_SelectLanguage(e_audio, "English");
1263     f_main_ut_UseService("TvChannel");
1264     f_main_ut_CheckLanguage("Programme", e_audio,
"English");
1265
1266     f_main_ut_SelectLanguage(e_audio, "German");
1267     f_main_ut_UseService("TvChannel");
1268     f_main_ut_CheckLanguage("Programme", e_audio,
"German");
1269
1270     // postamble
1271     f_main_ut_PowerOff();
1272     f_main_data_stopStreaming(0);
1273     f_main_data_stopStreaming(1);
1274     f_main_data_stopStreaming(2);
1275
1276     f_cf_Broadcast_down();
1277     f_cf_UpperTester_down();
1278 } // end test case TC_BCAST_ESG_CONF_107
1279 // change 39 (WK18): testcase description extension
with {extension "Description: 5.2.7 Service Guide with
dual audio streams via broadcast channel"}
1280
1281 /**
1282  *
1283  * @desc
1284  * <p>Service guide with purchase information</p>
1285  * <p>Test Case Description</p>
1286  * <p>with a terminal listening to the broadcast
Service Guide Delivery Channel.</p>
1287  * <p>when the terminal receives Service Guide
containing purchase item, purchase channel and purchase data fragments</p>
1288  * <p>then the terminal presents the information of
the Service Guide</p>
1289  * <p>Specification Reference</p>
1290  * <p>[BCAST10-ESG] Section 5.1.2.6, 5.1.2.7,
5.1.2.8, 5.5.1, 6.1.1<br>[BCAST10-sServices]</p>
1291  * <p>Preconditions</p>
```

```

1292      *      <p>Service guide cache of terminal is erased</p>
1293      *      <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
1294      *      <p>Access point information for service guide
entry point is configured in test tool</p>
1295      *      <p>This test cases uses the following SG fragment
instantiations found in the Appendix:
1296      *      <ul>
1297      *      <li>Service fragment with
Name="PayTvChannel"</li>
1298      *      <li>Content fragment with
Name="Programme", and StartTime and EndTime elements indicating values after
the time of test</li>
1299      *      <li>Schedule fragment for content fragment
with same values for startTime and endTime in the presentationWindow element</
li>
1300      *      <li>Access fragment for schedule fragement
with BroadcastServiceDelivery, KeyManagement, and EncryptionType elements and a
reference to the session description fragment</li>
1301      *      <li>Session Description fragment
containing SDP for "Programme"</li>
1302      *      <li>PurchaseItem fragment for service
fragment</li>
1303      *      <li>PurchaseData fragment for purchaseItem
fragment with PriceInfo and Description "Discount" elements</li>
1304      *      <li>PurchaseChannel fragment for
purchaseItem fragment with PurchaseURL pointing to a subscription site</li>
1305      *      </ul>
1306      *      </p>
1307      *      <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
1308      *      <p>Test Procedure</p>
1309      *      <p>
1310      *      <ul>
1311      *      <li>Set up the test tool to produce BCAST
service guide announcement and delivery using broadcast channel.</li>
1312      *      <li>Activate the BCAST application of the
terminal</li>
1313      *      <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
1314      *      <li>Browse the SG in the terminal</li>
1315      *      </ul>
1316      *      </p>
1317      *      <p>Pass-Criteria</p>
1318      *      <p>The following should be visible on terminal to
the end user after the delivery of the SG
1319      *      <ul>
1320      *      <li>The service "PayTvChannel" is
displayed with "Programme" schedule from start to end time, available as
"Discount" purchase, and price information </li>
1321      *      </ul>
1322      *      </p>
1323      *      @verdict
1324      *      pass in case the client pass the conformance test.
1325      *      @verdict
1326      *      fail in case the client does not pass the
conformance test.
1327      *      @verdict
1328      *      inconc in case a guard timer expires.
1329      */
1330      testcase TC_BCAST_ESG_CONF_108() runs on
BCastMainComponent system BCastPlatformComponent {
1331      // init components and ports
1332      f_createComponents();
1333      f_cf_Broadcast_up();
1334      f_cf_UpperTester_up();

```

```

1292      *      <p>Service guide cache of terminal is erased</p>
1293      *      <p>Terminal is configured to listen to BCAST
service guide announcements and delivery on the broadcast channel</p>
1294      *      <p>Access point information for service guide
entry point is configured in test tool</p>
1295      *      <p>This test cases uses the following SG fragment
instantiations found in the Appendix:
1296      *      <ul>
1297      *      <li>Service fragment with
Name="PayTvChannel"</li>
1298      *      <li>Content fragment with
Name="Programme", and StartTime and EndTime elements indicating values after
the time of test</li>
1299      *      <li>Schedule fragment for content fragment
with same values for startTime and endTime in the presentationWindow element</
li>
1300      *      <li>Access fragment for schedule fragement
with BroadcastServiceDelivery, KeyManagement, and EncryptionType elements and a
reference to the session description fragment</li>
1301      *      <li>Session Description fragment
containing SDP for "Programme"</li>
1302      *      <li>PurchaseItem fragment for service
fragment</li>
1303      *      <li>PurchaseData fragment for purchaseItem
fragment with PriceInfo and Description "Discount" elements</li>
1304      *      <li>PurchaseChannel fragment for
purchaseItem fragment with PurchaseURL pointing to a subscription site</li>
1305      *      </ul>
1306      *      </p>
1307      *      <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery.</p>
1308      *      <p>Test Procedure</p>
1309      *      <p>
1310      *      <ul>
1311      *      <li>Set up the test tool to produce BCAST
service guide announcement and delivery using broadcast channel.</li>
1312      *      <li>Activate the BCAST application of the
terminal</li>
1313      *      <li>Request from BCAST application on
terminal to get the service guide (optional)</li>
1314      *      <li>Browse the SG in the terminal</li>
1315      *      </ul>
1316      *      </p>
1317      *      <p>Pass-Criteria</p>
1318      *      <p>The following should be visible on terminal to
the end user after the delivery of the SG
1319      *      <ul>
1320      *      <li>The service "PayTvChannel" is
displayed with "Programme" schedule from start to end time, available as
"Discount" purchase, and price information </li>
1321      *      </ul>
1322      *      </p>
1323      *      @verdict
1324      *      pass in case the client pass the conformance test.
1325      *      @verdict
1326      *      fail in case the client does not pass the
conformance test.
1327      *      @verdict
1328      *      inconc in case a guard timer expires.
1329      */
1330      testcase TC_BCAST_ESG_CONF_108() runs on
BCastMainComponent system BCastPlatformComponent {
1331      // init components and ports
1332      f_createComponents();
1333      f_cf_Broadcast_up();
1334      f_cf_UpperTester_up();

```

```
1335
1336                // preamble
1337                f_startCommonUtPreamble();
1338
1339                // test behavior
1340                var ServiceGuideDeliveryUnit v_SGDU := {};
1341                var UInt32 v_FragmentVersion := 1;
1342
1343                f_main_ctrl_InitStartTime(c_one_hour,
c_one_hour);
1344                // change 01 (WK 6): global time
1345                definition
1346                var DateTime v_DateTime_StartTime :=
1347                f_getDateTime(v_startTime);
1348                // change 01 (WK 6): global time
1349                definition
1350                var DateTime v_DateTime_EndTime := f_getDateTime
1351                (v_endTime);
1352                // change 01 (WK 6): global time definition
1353                var SdpFragment v_Sdp := valueof
1354                (m_def_SdpFragment(
1355                PX_SDP_VIDEO_PROG_1_REF_ID,
1356                PX_SDP_VIDEO_PROG_1
1357                ));
1358                var AccessTypeType v_AccessType := valueof
1359                (m_def_AccessType_bc_sdp(
1360                f_getAccessType
1361                (PX_BROADCAST_NETWORK_BEARER),
1362                PX_SDP_VIDEO_PROG_1_REF_ID
1363                ));
1364                var PriceInfoType v_PriceInfo := valueof
1365                (m_PriceInfo(
1366                1,
1367                PX_MONETARY_PRICE,
1368                PX_CURRENCY
1369                ));
1370                var ServiceType v_Service := valueof
1371                (m_def_Service(
1372                PX_SGDU_SERVICE_ID,
1373                v_FragmentVersion,
1374                "PayTvChannel",
1375                c_basicTv_ServiceType
1376                ));
1377                var ContentType v_Content := valueof
1378                (m_def_Content(
1379                PX_SGDU_CONTENT_ID_PROG_1,
1380                v_FragmentVersion,
1381                "Programme",
1382                PX_SGDU_SERVICE_ID,
1383                v_DateTime_StartTime,
1384                v_DateTime_EndTime
1385                ));
1386                var ScheduleType v_Schedule := valueof
1387                (m_def_Schedule(
1388                PX_SGDU_SCHEDULE_ID_PROG_1,
1389                v_FragmentVersion,
1390                PX_SGDU_SERVICE_ID,
1391                PX_SGDU_CONTENT_ID_PROG_1,
1392                v_startTime,
1393                // change 01 (WK 6):
1394                global time definition
```

```
1335
1336                // preamble
1337                f_startCommonUtPreamble();
1338
1339                // test behavior
1340                var ServiceGuideDeliveryUnit v_SGDU := {};
1341                var UInt32 v_FragmentVersion := 1;
1342
1343                f_main_ctrl_InitStartTime(c_one_hour,
c_one_hour);
1344                // change 01 (WK 6): global time
1345                definition
1346                var DateTime v_DateTime_StartTime :=
1347                f_getDateTime(v_startTime);
1348                // change 01 (WK 6): global time
1349                definition
1350                var DateTime v_DateTime_EndTime := f_getDateTime
1351                (v_endTime);
1352                // change 01 (WK 6): global time definition
1353                var SdpFragment v_Sdp := valueof
1354                (m_def_SdpFragment(
1355                PX_SDP_VIDEO_PROG_1_REF_ID,
1356                PX_SDP_VIDEO_PROG_1
1357                ));
1358                var AccessTypeType v_AccessType := valueof
1359                (m_def_AccessType_bc_sdp(
1360                f_getAccessType
1361                (PX_BROADCAST_NETWORK_BEARER),
1362                PX_SDP_VIDEO_PROG_1_REF_ID
1363                ));
1364                var PriceInfoType v_PriceInfo := valueof
1365                (m_PriceInfo(
1366                1,
1367                PX_MONETARY_PRICE,
1368                PX_CURRENCY
1369                ));
1370                var ServiceType v_Service := valueof
1371                (m_def_Service(
1372                PX_SGDU_SERVICE_ID,
1373                v_FragmentVersion,
1374                "PayTvChannel",
1375                c_basicTv_ServiceType
1376                ));
1377                var ContentType v_Content := valueof
1378                (m_def_Content(
1379                PX_SGDU_CONTENT_ID_PROG_1,
1380                v_FragmentVersion,
1381                "Programme",
1382                PX_SGDU_SERVICE_ID,
1383                v_DateTime_StartTime,
1384                v_DateTime_EndTime
1385                ));
1386                var ScheduleType v_Schedule := valueof
1387                (m_def_Schedule(
1388                PX_SGDU_SCHEDULE_ID_PROG_1,
1389                v_FragmentVersion,
1390                PX_SGDU_SERVICE_ID,
1391                PX_SGDU_CONTENT_ID_PROG_1,
1392                v_startTime,
1393                // change 01 (WK 6):
1394                global time definition
```

```
1386 v_endTime // change 01 (WK 6):
1387 global time definition
1388
1389 // TODO: Omitted here is setting of KMS value
1390 (see open issues) but omission may be ok in this test case
1391 var AccessType v_Access := valueof(m_def_Access(
1392     PX_SGDU_ACCESS_ID_PROG_1,
1393     v_FragmentVersion,
1394     v_AccessType,
1395     PX_SGDU_SERVICE_ID,
1396     PX_SGDU_SCHEDULE_ID_PROG_1,
1397     c_class_SG
1398 ));
1399 var PurchaseItemType v_PurchaseItem := valueof
1400 (m_def_PurchaseItem(
1401     PX_SGDU_PURCHASE_ITEM_ID,
1402     v_FragmentVersion,
1403     PX_SGDU_PURCHASE_DATA_ID,
1404     PX_SGDU_SERVICE_ID,
1405     "PurchaseItem1"
1406 ));
1407 var PurchaseChannelType v_PurchaseChannel :=
1408 valueof(m_def_PurchaseChannel(
1409     PX_SGDU_PURCHASE_CHANNEL_ID,
1410     v_FragmentVersion,
1411     "PurchaseChannel",
1412     PX_PURCHASE_URL
1413 ));
1414 var PurchaseDataType v_PurchaseData := valueof
1415 (m_def_PurchaseData(
1416     PX_SGDU_PURCHASE_DATA_ID,
1417     v_FragmentVersion,
1418     {{omit,"Discount"}}},
1419     PX_SGDU_PURCHASE_ITEM_ID,
1420     PX_SGDU_PURCHASE_CHANNEL_ID,
1421     v_PriceInfo
1422 ));
1423 f_addServiceFragment(v_SGDU, v_Service);
1424 f_addContentFragment(v_SGDU, v_Content);
1425 f_addScheduleFragment(v_SGDU, v_Schedule);
1426 f_addAccessFragment(v_SGDU, v_Access);
1427 f_addPurchaseItemFragment(v_SGDU,
1428 v_PurchaseItem);
1429 f_addPurchaseChannelFragment(v_SGDU,
1430 v_PurchaseChannel);
1431 f_addPurchaseDataFragment(v_SGDU,
1432 v_PurchaseData);
1433 f_addSdpFragment(v_SGDU, v_FragmentVersion,
1434 v_Sdp);
1435 f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
1436 f_main_ut_RunBCastApplication();
1437 if (PX_GET_SG_DISPLAY_MANUALLY) {
1438     f_main_ut_GetServiceGuide(PX_SGDD_ID);
1439 }
1440 f_main_ut_CheckService("PayTvChannel");
1441 f_main_ut_SelectService("PayTvChannel");
1442 f_main_ut_CheckContent("Programme",
1443 v_DateTime_StartTime, v_DateTime_EndTime);
```

```
1386 v_endTime // change 01 (WK 6):
1387 global time definition
1388
1389 // TODO: Omitted here is setting of KMS value
1390 (see open issues) but omission may be ok in this test case
1391 var AccessType v_Access := valueof(m_def_Access(
1392     PX_SGDU_ACCESS_ID_PROG_1,
1393     v_FragmentVersion,
1394     v_AccessType,
1395     PX_SGDU_SERVICE_ID,
1396     PX_SGDU_SCHEDULE_ID_PROG_1,
1397     c_class_SG
1398 ));
1399 var PurchaseItemType v_PurchaseItem := valueof
1400 (m_def_PurchaseItem(
1401     PX_SGDU_PURCHASE_ITEM_ID,
1402     v_FragmentVersion,
1403     PX_SGDU_PURCHASE_DATA_ID,
1404     PX_SGDU_SERVICE_ID,
1405     "PurchaseItem1"
1406 ));
1407 var PurchaseChannelType v_PurchaseChannel :=
1408 valueof(m_def_PurchaseChannel(
1409     PX_SGDU_PURCHASE_CHANNEL_ID,
1410     v_FragmentVersion,
1411     "PurchaseChannel",
1412     PX_PURCHASE_URL
1413 ));
1414 var PurchaseDataType v_PurchaseData := valueof
1415 (m_def_PurchaseData(
1416     PX_SGDU_PURCHASE_DATA_ID,
1417     v_FragmentVersion,
1418     {{omit,"Discount"}}},
1419     PX_SGDU_PURCHASE_ITEM_ID,
1420     PX_SGDU_PURCHASE_CHANNEL_ID,
1421     v_PriceInfo
1422 ));
1423 f_addServiceFragment(v_SGDU, v_Service);
1424 f_addContentFragment(v_SGDU, v_Content);
1425 f_addScheduleFragment(v_SGDU, v_Schedule);
1426 f_addAccessFragment(v_SGDU, v_Access);
1427 f_addPurchaseItemFragment(v_SGDU,
1428 v_PurchaseItem);
1429 f_addPurchaseChannelFragment(v_SGDU,
1430 v_PurchaseChannel);
1431 f_addPurchaseDataFragment(v_SGDU,
1432 v_PurchaseData);
1433 f_addSdpFragment(v_SGDU, v_FragmentVersion,
1434 v_Sdp);
1435 f_main_ctrl_broadcastServiceGuide(v_SGDU, omit);
1436 f_main_ut_RunBCastApplication();
1437 if (PX_GET_SG_DISPLAY_MANUALLY) {
1438     f_main_ut_GetServiceGuide(PX_SGDD_ID);
1439 }
1440 f_main_ut_CheckService("PayTvChannel");
1441 f_main_ut_SelectService("PayTvChannel");
1442 f_main_ut_CheckContent("Programme",
1443 v_DateTime_StartTime, v_DateTime_EndTime);
```

```
1441         f_main_ut_CheckPurchaseInfo( "Discount" );
1442
1443         // postamble
1444         f_main_ut_PowerOff();
1445
1446         f_cf_Broadcast_down();
1447         f_cf_UpperTester_down();
1448     }// end test case TC_BCAST_ESG_CONF_108
1449     // change 39 (WK18): testcase description extension
1450     with {extension "Description: 5.2.8 Service Guide
update via interaction channel (same fragment id, higher version number)"}

```

```
1441         f_main_ut_CheckPurchaseInfo( "Discount" );
1442
1443         // postamble
1444         f_main_ut_PowerOff();
1445
1446         f_cf_Broadcast_down();
1447         f_cf_UpperTester_down();
1448     }// end test case TC_BCAST_ESG_CONF_108
1449     // change 39 (WK18): testcase description extension
1450     with {extension "Description: 5.2.8 Service Guide
update via interaction channel (same fragment id, higher version number)"}

```

```
1451     /**
1452     * @desc
1453     * <p>Service Guide update via interaction channel
1454     (same fragment id, lower version number)</p>
1455     * <p>Test Case Description</p>
1456     * <p>with a terminal having requested and received
a Service Guide over the interactive Delivery Channel.</p>
1457     * <p>when the terminal receives Service Guide
update containing a content fragment with same id and lower version</p>
1458     * <p>then the terminal does not present the
information of the updated Service Guide</p>
1459     * <p>Specification Reference</p>
1460     * <p>[BCAST10 -ESG] Section 5.4.2.1.2, 5.5 4th
bullet item</p>
1461     * <p>Preconditions</p>
1462     * <p>Service guide cache of terminal is erased</p>
1463     * <p>Terminal is configured to request BCAST
service guide delivery on the interactive channel</p>
1464     * <p>Access point information for service guide
entry point is configured in test tool</p>
1465     * <p>This test case uses the following SG fragment
instantiations:</p>
1466     * <p>Initial service guide:
1467     * <ul>
1468     * <li>Service fragment with
Name="TvChannel"</li>
1469     * <li>Content fragment with version = "2"
and Name = "Programm1-version2" , and StartTime and EndTime elements
indicating values after the time of test</li>
1470     * <li>Schedule fragment for "Programm1-
version2" content fragment with same values for startTime and endTime in the
presentationWindow element</li>
1471     * <li>Access fragment for "Programm1-
version2" schedule fragment</li>
1472     * </ul>
1473     * <p>
1474     * <p>Updated service guide:
1475     * <ul>
1476     * <li>Service fragment with
Name="TvChannel"</li>
1477     * <li>Content fragment with version = "1"
and Name = "Programm1" , and StartTime and EndTime elements indicating values
after the time of test</li>
1478     * <li>Schedule fragment for "Programm1"
content fragment with same values for startTime and endTime in the
presentationWindow element</li>
1479     * <li>Access fragment for "Programm1"
schedule fragment</li>
1480     * </ul>
1481     * <p>
1482     * <p>Note: All the fragments are associated with
the same Service fragment and are sent in the same service guide delivery</p>
1483     * <p>Test Procedure</p>
1484     * <p>

```

<>

```

1485         *      <ul>
1486         *      <li>Set up the test tool to enable initial
BCAST service guide delivery using interactive channel.</li>
1487         *      <li>Activate the BCAST application of the
terminal</li>
1488         *      <li>Request from BCAST application on
terminal to get the service guide</li>
1489         *      <li>Browse the SG on the terminal</li>
1490         *      <li>Set up the test tool to make an
updated BCAST service guide available to the terminal using the interactive
channel.</li>
1491         *      <li>Request from BCAST application on
terminal to update the service guide .</li>
1492         *      <li>Browse the SG on the terminal</li>
1493         *      </ul>
1494         *      </p>
1495         *      <p>Pass-Criteria</p>
1496         *      <p>The following should be visible to the end
user after the initial delivery of the SG
1497         *      <ul>
1498         *      <li>There is a service called "TvChannel"
associated with program "Programm1-version2" as well as start and end time
values(There is a "TvChannel" that contains "Programm1-version2" scheduled
from startTime to endTime</li>
1499         *      </ul>
1500         *      </p>
1501         *      <p>The following should be visible to the end
user after the delivery of the updated SG
1502         *      <ul>
1503         *      <li>There is a service called "TvChannel"
associated with program "Programm1-version2" as well as start and end time
values(There is a "TvChannel" that contains "Programm1-version2" scheduled
from startTime to endTime)).</li>
1504         *      </ul>
1505         *      </p>
1506         *      @verdict
1507         *      pass in case the client pass the conformance test.
1508         *      @verdict
1509         *      fail in case the client does not pass the
conformance test.
1510         *      @verdict
1511         *      inconc in case a guard timer expires.
1512         */
1513         testcase TC_BCAST_ESG_CONF_109() runs on
BCastMainComponent system BCastPlatformComponent {
1514             // init components and ports
1515             f_createComponents();
1516             f_cf_Interaction_up();
1517             f_cf_UpperTester_up();
1518
1519             // preamble
1520             f_startCommonUtPreamble();
1521
1522             var ServiceGuideDeliveryUnit v_SGDU := {};
1523             var UInt32 v_FragmentVersion := 1;
1524
1525             var UInt v_NTP_StartTime := 0;
1526             var UInt v_NTP_EndTime := 0;
1527             f_InitStartEndTime(v_NTP_StartTime,
v_NTP_EndTime);
1528
1529             var DateTime v_DateTime_StartTime :=
f_getDateTime(v_NTP_StartTime);
1530             var DateTime v_DateTime_EndTime := f_getDateTime
(v_NTP_EndTime);
1531

```



```

1532         var SdpFragment v_Sdp := valueof
1533         (m_def_SdpFragment(
1534             PX_SDP_VIDEO_PROG_1_REF_ID,
1535             PX_SDP_VIDEO_PROG_1
1536         ));
1537         var AccessTypeType v_AccessType := valueof
1538         (m_def_AccessType_bc_sdp(
1539             f_getAccessType
1540             (PX_INTERACTION_NETWORK_BEARER),
1541             PX_SDP_VIDEO_PROG_1_REF_ID
1542         ));
1543         var ServiceType v_Service := valueof
1544         (m_def_Service(
1545             PX_SGDU_SERVICE_ID,
1546             v_FragmentVersion,
1547             "TvChannel",
1548             c_basicTv_ServiceType
1549         ));
1550         var ContentType v_Content := valueof
1551         (m_def_Content(
1552             PX_SGDU_CONTENT_ID_PROG_1,
1553             f_IncFragmentVersion(v_FragmentVersion),
1554             "Programm1-version2",
1555             PX_SGDU_SERVICE_ID,
1556             v_DateTime_StartTime,
1557             v_DateTime_EndTime
1558         ));
1559         var ScheduleType v_Schedule := valueof
1560         (m_def_Schedule(
1561             PX_SGDU_SCHEDULE_ID_PROG_1,
1562             v_FragmentVersion,
1563             PX_SGDU_SERVICE_ID,
1564             PX_SGDU_CONTENT_ID_PROG_1,
1565             v_NTP_StartTime,
1566             v_NTP_EndTime
1567         ));
1568         var AccessType v_Access := valueof(m_def_Access(
1569             PX_SGDU_ACCESS_ID_PROG_1,
1570             v_FragmentVersion,
1571             v_AccessType,
1572             PX_SGDU_SERVICE_ID,
1573             PX_SGDU_SCHEDULE_ID_PROG_1,
1574             c_class_SG
1575         ));
1576         f_addServiceFragment(v_SGDU, v_Service);
1577         f_addContentFragment(v_SGDU, v_Content);
1578         f_addScheduleFragment(v_SGDU, v_Schedule);
1579         f_addAccessFragment(v_SGDU, v_Access);
1580         f_addSdpFragment(v_SGDU, v_FragmentVersion,
1581         v_Sdp);
1582         f_main_ut_RunBCastApplication();
1583         if (PX_GET_SG_DISPLAY_MANUALLY) {
1584             f_main_ut_GetServiceGuide
1585             (PX_SGDD_ID);
1586         }
1587         f_main_ctrl_ServiceGuideOnRequest(v_SGDU);
1588         f_main_ut_CheckService("TvChannel");
1589         f_main_ut_SelectService("TvChannel");

```

| | | | | |
|------|--|-----|------|---|
| 1590 | f_main_ut_CheckContent("Programm1-version2", v_DateTime_StartTime, v_DateTime_EndTime); | | | |
| 1591 | | | | |
| 1592 | // test behavior | | | |
| 1593 | v_SGDU := {}; | | | |
| 1594 | v_Content.version := v_FragmentVersion; | | | |
| 1595 | v_Content.Names := {{omit,"Programm1"}}; | | | |
| 1596 | f_addContentFragment(v_SGDU, v_Content); | | | |
| 1597 | | | | |
| 1598 | f_main_ctrl_ServiceGuideOnRequest(v_SGDU); | | | |
| 1599 | | | | |
| 1600 | if (PX_GET_SG_UPDATE_MANUALLY) { | | | |
| 1601 | f_main_ut_UpdateServiceGuide(); | | | |
| 1602 | } | | | |
| 1603 | | | | |
| 1604 | f_main_ut_CheckService("TvChannel"); | | | |
| 1605 | f_main_ut_SelectService("TvChannel"); | | | |
| 1606 | f_main_ut_CheckContent("Programm1-version2", v_DateTime_StartTime, v_DateTime_EndTime); | | | |
| 1607 | | | | |
| 1608 | // postamble | | | |
| 1609 | f_main_ut_PowerOff(); | | | |
| 1610 | | | | |
| 1611 | f_cf_Broadcast_down(); | | | |
| 1612 | f_cf_UpperTester_down(); | | | |
| 1613 | } // end test case TC_BCAST_ESG_CONF_109 | | | |
| 1614 | } // end group clientConformanceTestCases | = | 1451 | } // end group clientConformanceTestCases |
| 1615 | }// end group bcastConformanceTestCases | | 1452 | }// end group bcastConformanceTestCases |
| 1616 | } // end group clientConformanceTestCases | + - | | |
| 1617 | }// end group bcastConformanceTestCases | | | |
| 1618 | } | = | 1453 | } |