

```
126 "525": {
127     "mediaId": 525,
128     "ownerId": 10,
129     "parentId": 0,
130     "name": "TestRenban_4. jpg",
131     "mediaType": "image",
132     "storedAs": "525. jpg",
133     "fileName": "TestRenban_4. jpg",
134     "tags": null,
135     "tagValues": null,
136     "fileSize": "8660",
137     "duration": 15,
138     "valid": 1,
139     "moduleSystemFile": 0,
140     "expires": 0,
141     "retired": "0",
142     "isEdited": 0,
143     "md5": "46ed66b7d699d470818aeaf7b5cbec79",
144     "owner": "xibo_provider",
145     "groupsWithPermissions": null,
146     "released": 1,
147     "apiRef": null,
148     "createdDt": "2023-09-13 16:09:02",
149     "modifiedDt": "2023-09-13 16:09:02",
150     "enableStat": "Inherit",
151     "isSaveRequired": null,
152     "isRemote": null,
153     "cloned": false,
154     "newExpiry": null,
155     "alwaysCopy": false
156 },
157 "526": {
158     "mediaId": 526,
159     "ownerId": 10,
160     "parentId": 0,
161     "name": "TestRenban_5. jpg",
162     "mediaType": "image",
163     "storedAs": "526. jpg",
164     "fileName": "TestRenban_5. jpg",
165     "tags": null,
166     "tagValues": null,
167     "fileSize": "11530",
168     "duration": 15,
169     "valid": 1,
170     "moduleSystemFile": 0,
171     "expires": 0,
172     "retired": "0",
173     "isEdited": 0,
174     "md5": "9e4e1f1e96b723a2c48984eaa9f53036",
175     "owner": "xibo_provider",
176     "groupsWithPermissions": null,
177     "released": 1,
178     "apiRef": null,
179     "createdDt": "2023-09-13 16:09:03",
```

```
180     "modifiedDt": "2023-09-13 16:09:03",
181     "enableStat": "Inherit",
182     "isSaveRequired": null,
183     "isRemote": null,
184     "cloned": false,
185     "newExpiry": null,
186     "alwaysCopy": false
187 },
188 "527": {
189     "mediaId": 527,
190     "ownerId": 10,
191     "parentId": 0,
192     "name": "TestRenban_6.jpg",
193     "mediaType": "image",
194     "storedAs": "527.jpg",
195     "fileName": "TestRenban_6.jpg",
196     "tags": null,
197     "tagValues": null,
198     "fileSize": "9637",
199     "duration": 15,
200     "valid": 1,
201     "moduleSystemFile": 0,
202     "expires": 0,
203     "retired": "0",
204     "isEdited": 0,
205     "md5": "caf4c161d0916d284b77e2ed9fc256fe",
206     "owner": "xibo_provider",
207     "groupsWithPermissions": null,
208     "released": 1,
209     "apiRef": null,
210     "createdDt": "2023-09-13 16:09:03",
211     "modifiedDt": "2023-09-13 16:09:03",
212     "enableStat": "Inherit",
213     "isSaveRequired": null,
214     "isRemote": null,
215     "cloned": false,
216     "newExpiry": null,
217     "alwaysCopy": false
218 },
219 "528": {
220     "mediaId": 528,
221     "ownerId": 10,
222     "parentId": 0,
223     "name": "TestRenban_7.jpg",
224     "mediaType": "image",
225     "storedAs": "528.jpg",
226     "fileName": "TestRenban_7.jpg",
227     "tags": null,
228     "tagValues": null,
229     "fileSize": "8575",
230     "duration": 15,
231     "valid": 1,
232     "moduleSystemFile": 0,
233     "expires": 0,
```

```
234     "retired": "0",
235     "isEdited": 0,
236     "md5": "f7cda477ff3d9f3be56a8925b494fb98",
237     "owner": "xibo_provider",
238     "groupsWithPermissions": null,
239     "released": 1,
240     "apiRef": null,
241     "createdDt": "2023-09-13 16:09:03",
242     "modifiedDt": "2023-09-13 16:09:03",
243     "enableStat": "Inherit",
244     "isSaveRequired": null,
245     "isRemote": null,
246     "cloned": false,
247     "newExpiry": null,
248     "alwaysCopy": false
249 },
250 "529": {
251     "mediaId": 529,
252     "ownerId": 10,
253     "parentId": 0,
254     "name": "TestRenban_8. jpg",
255     "mediaType": "image",
256     "storedAs": "529. jpg",
257     "fileName": "TestRenban_8. jpg",
258     "tags": null,
259     "tagValues": null,
260     "fileSize": "10206",
261     "duration": 15,
262     "valid": 1,
263     "moduleSystemFile": 0,
264     "expires": 0,
265     "retired": "0",
266     "isEdited": 0,
267     "md5": "60bb3e5484733396ed89726b06ff25ab",
268     "owner": "xibo_provider",
269     "groupsWithPermissions": null,
270     "released": 1,
271     "apiRef": null,
272     "createdDt": "2023-09-13 16:09:04",
273     "modifiedDt": "2023-09-13 16:09:04",
274     "enableStat": "Inherit",
275     "isSaveRequired": null,
276     "isRemote": null,
277     "cloned": false,
278     "newExpiry": null,
279     "alwaysCopy": false
280 },
281 "530": {
282     "mediaId": 530,
283     "ownerId": 10,
284     "parentId": 0,
285     "name": "TestRenban_9. jpg",
286     "mediaType": "image",
287     "storedAs": "530. jpg",
```

```

288     "fileName": "TestRenban_9. jpg",
289     "tags": null,
290     "tagValues": null,
291     "fileSize": "10662",
292     "duration": 15,
293     "valid": 1,
294     "moduleSystemFile": 0,
295     "expires": 0,
296     "retired": "0",
297     "isEdited": 0,
298     "md5": "c1c930a9e80dfa2a465bf04faa36ce3f",
299     "owner": "xibo_provider",
300     "groupsWithPermissions": null,
301     "released": 1,
302     "apiRef": null,
303     "createdDt": "2023-09-13 16:09:04",
304     "modifiedDt": "2023-09-13 16:09:04",
305     "enableStat": "Inherit",
306     "isSaveRequired": null,
307     "isRemote": null,
308     "cloned": false,
309     "newExpiry": null,
310     "alwaysCopy": false
311 }
312 }

```

However, unfortunately, this synchronization task is not 0-9, but 0142563789.

To investigate this, let's create a query that will give us the current realistic execution order.

```

1 SELECT
2     m.name AS '画像名', w.*
3 FROM
4     xibo_xibo233.media m
5 LEFT JOIN
6     xibo_xibo233.lkwidgetmedia lkwm
7     ON lkwm.mediaId=m.mediaId
8 LEFT JOIN
9     xibo_xibo233.widget w
10    ON w.widgetId=lkwm.widgetId
11 WHERE 1=1
12     AND m.type LIKE "image"
13     AND m.name LIKE "TestRenban%"
14     AND w.playlistId=343
15 ORDER BY
16     w.displayOrder

```

画像名	widgetId	playlistId	ownerId	type	duration	displayOrder	calculatedDuration	useDuration	fromDt	toDt	createdDt	modifiedDt
TestRenban_0.jpg	1057	343	10	image	15	1	15	1	0	2147483647	1694568302	1694568302
TestRenban_1.jpg	1058	343	10	image	15	2	15	1	0	2147483647	1694568302	1694568302
TestRenban_5.jpg	1062	343	10	image	15	3	15	1	0	2147483647	1694568302	1694576942
TestRenban_6.jpg	1063	343	10	image	15	4	15	1	0	2147483647	1694568302	1694576942
TestRenban_7.jpg	1064	343	10	image	15	5	15	1	0	2147483647	1694568302	1694576942
TestRenban_2.jpg	1070	343	10	image	15	6	15	1	0	2147483647	1694576942	1694576942
TestRenban_8.jpg	1065	343	10	image	15	7	15	1	0	2147483647	1694568302	1694576942
TestRenban_3.jpg	1071	343	10	image	15	8	15	1	0	2147483647	1694576942	1694576942
TestRenban_9.jpg	1066	343	10	image	15	9	15	1	0	2147483647	1694568302	1694576942
TestRenban_4.jpg	1072	343	10	image	15	10	15	1	0	2147483647	1694576942	1694576942

Ten actual display orders of mediaId 520 to 530 are obtained.

ID	名前	タイプ	サムネイル	期間	サイズ	オーナー	権限	ファイル名	統計	登録日	期限切れ	行メニュー
530	TestRenban_9.jpg	image	9	0:00:15	10.41 KIB	xibo_provider		TestRenban_9.jpg	1	2023-09-13 16:09	利用しない	
529	TestRenban_8.jpg	image	8	0:00:15	9.97 KIB	xibo_provider		TestRenban_8.jpg	1	2023-09-13 16:09	利用しない	
528	TestRenban_7.jpg	image	7	0:00:15	8.37 KIB	xibo_provider		TestRenban_7.jpg	1	2023-09-13 16:09	利用しない	
527	TestRenban_6.jpg	image	6	0:00:15	9.41 KIB	xibo_provider		TestRenban_6.jpg	1	2023-09-13 16:09	利用しない	
526	TestRenban_5.jpg	image	5	0:00:15	11.26 KIB	xibo_provider		TestRenban_5.jpg	1	2023-09-13 16:09	利用しない	
525	TestRenban_4.jpg	image	4	0:00:15	8.46 KIB	xibo_provider		TestRenban_4.jpg	1	2023-09-13 16:09	利用しない	
523	TestRenban_3.jpg	image	3	0:00:15	13.78 KIB	xibo_provider		TestRenban_3.jpg	1	2023-09-13 15:49	利用しない	
522	TestRenban_2.jpg	image	2	0:00:15	11.62 KIB	xibo_provider		TestRenban_2.jpg	1	2023-09-13 15:49	利用しない	
521	TestRenban_1.jpg	image	1	0:00:15	7.02 KIB	xibo_provider		TestRenban_1.jpg	1	2023-09-13 15:49	利用しない	
520	TestRenban_0.jpg	image	0	0:00:15	12.39 KIB	xibo_provider		TestRenban_0.jpg	1	2023-09-13 15:49	利用しない	

Therefore, the array may not be processed in order.

In other words, there is a possibility that CreateAndAssign is doing something wrong.

```

error_log(json_encode($media));

// Do we have any mediaId's left which should be assigned and aren't?
// Add the ones we have left
foreach ($media as $item) {
    if (in_array($item->mediaId, $mediaIds)) {
        $assignmentMade = true;
        $this->createAndAssign($playlist, $item, $count);
    }
}

```

```
DynamicPlaylistSyncTask.php × { } 番号が狂ってる状態.json
> Users > MIC128~1.KK- > AppData > Local > Temp > scp43184 > var > www > xibo233 > admin > lib > XTR > DynamicPlaylistSyncTask.php > createAndAssign Aa _ab_* ? of 2
247 * @throws NotFoundException
248 */
249 private function createAndAssign($playlist, $media, $displayOrder)
250 {
251     $this->log->debug('Media Item needs to be assigned ' . $media->name . ' in sequence ' . $displayOrder);
252
253     // Create a module
254     $module = $this->moduleFactory->create($media->mediaType);
255
256     if ($module->getModule()->assignable == 1) {
257
258         // Determine the duration
259         $mediaDuration = ($media->duration == 0) ? $module->determineDuration() : $media->duration;
260
261         // Create a widget
262         $widget = $this->widgetFactory->create(
263             $playlist->getOwnerId(),
264             $playlist->playlistId,
265             $media->mediaType,
266             $mediaDuration
267         );
268
269         $widget->assignMedia($media->mediaId);
270         $widget->displayOrder = $displayOrder;
271
272         // Assign the widget to the module
273         $module->setWidget($widget);
274
275         // Set default options (this sets options on the widget)
276         $module->setDefaultWidgetOptions();
277
278         // Set the duration to be equal to the Library duration of this media
279         $module->widget->useDuration = 1;
280         $module->widget->duration = $mediaDuration;
281         $module->widget->calculatedDuration = $mediaDuration;
282
283         // Assign the widget to the playlist
284         $playlist->assignWidget($widget);
285     }
286 }
ターミナル 問題 出力 デバッグコンソール SQL CONSOLE
```

I added an error log

```
244 */
245 private function createAndAssign($playlist, $media, $displayOrder)
246 {
247     $this->log->debug('Media Item needs to be assigned ' . $media->name . ' in sequence ' . $displayOrder);
248
249     //検証用
250     if($playlist->name == "No2 2023/09/01 16:17 cronタスクのテスト用 TestRenban_"){
251         error_log("No2 2023/09/01 16:17 cronタスクのテスト用 TestRenban_");
252         error_log(json_encode($playlist));
253         error_log(json_encode($media));
254         error_log("表示順序 : ".$displayOrder);
255     }
256 }
```

Try throwing in ClearAll and 0-9.

In this case, we have confirmed that the numbers are the same, and this time they are as follows.

画像名	widgetId	playlistId	ownerId	type	duration	displayOrder	calculatedDuration	useDuration	fromDt	toDt	createdDt	modifiedDt	mediaId	name
TestRenban_0.jpg	1169	343	10	image	15	1	15	1	0	2147483647	1694594883	1694594883	543	TestRenban_0.jpg
TestRenban_1.jpg	1170	343	10	image	15	2	15	1	0	2147483647	1694594883	1694594883	544	TestRenban_1.jpg
TestRenban_2.jpg	1171	343	10	image	15	3	15	1	0	2147483647	1694594883	1694594883	545	TestRenban_2.jpg
TestRenban_3.jpg	1172	343	10	image	15	4	15	1	0	2147483647	1694594883	1694594883	546	TestRenban_3.jpg
TestRenban_4.jpg	1173	343	10	image	15	5	15	1	0	2147483647	1694594883	1694594883	547	TestRenban_4.jpg
TestRenban_5.jpg	1174	343	10	image	15	6	15	1	0	2147483647	1694594883	1694594883	548	TestRenban_5.jpg
TestRenban_6.jpg	1175	343	10	image	15	7	15	1	0	2147483647	1694594883	1694594883	549	TestRenban_6.jpg
TestRenban_7.jpg	1176	343	10	image	15	8	15	1	0	2147483647	1694594883	1694594883	550	TestRenban_7.jpg
TestRenban_8.jpg	1177	343	10	image	15	9	15	1	0	2147483647	1694594883	1694594883	551	TestRenban_8.jpg
TestRenban_9.jpg	1178	343	10	image	15	10	15	1	0	2147483647	1694594883	1694594883	552	TestRenban_9.jpg

Then, save the resulting debug log (it is recommended to keep it closed as there are about 2500 lines...) as a "json with the same numbers".

► The state in which the numbers are aligned.json

Let's analyze the contents

First, "displayOrder". This is a string contained in \$playlist.

1	102:	"displayOrder": 1,
2	219:	"displayOrder": 1,
3	259:	"displayOrder": 2,
4	376:	"displayOrder": 1,
5	416:	"displayOrder": 2,
6	456:	"displayOrder": 3,
7	573:	"displayOrder": 1,
8	613:	"displayOrder": 2,
9	653:	"displayOrder": 3,
10	693:	"displayOrder": 4,
11	810:	"displayOrder": 1,
12	850:	"displayOrder": 2,
13	890:	"displayOrder": 3,
14	930:	"displayOrder": 4,
15	970:	"displayOrder": 5,
16	1087:	"displayOrder": 1,
17	1127:	"displayOrder": 2,
18	1167:	"displayOrder": 3,
19	1207:	"displayOrder": 4,
20	1247:	"displayOrder": 5,
21	1287:	"displayOrder": 6,
22	1404:	"displayOrder": 1,
23	1444:	"displayOrder": 2,
24	1484:	"displayOrder": 3,
25	1524:	"displayOrder": 4,
26	1564:	"displayOrder": 5,
27	1604:	"displayOrder": 6,
28	1644:	"displayOrder": 7,
29	1761:	"displayOrder": 1,
30	1801:	"displayOrder": 2,
31	1841:	"displayOrder": 3,
32	1881:	"displayOrder": 4,
33	1921:	"displayOrder": 5,
34	1961:	"displayOrder": 6,
35	2001:	"displayOrder": 7,
36	2041:	"displayOrder": 8,

```
37 2158:         "displayOrder": 1,
38 2198:         "displayOrder": 2,
39 2238:         "displayOrder": 3,
40 2278:         "displayOrder": 4,
41 2318:         "displayOrder": 5,
42 2358:         "displayOrder": 6,
43 2398:         "displayOrder": 7,
44 2438:         "displayOrder": 8,
45 2478:         "displayOrder": 9,
```

It can be seen that displayOrder increases by 1 (the second order is added in this loop)

12  
123

...

123456789 (the tenth order is added in this loop) . This alone does not provide the actual content order information. Therefore, since \$playlist has a string called mediaIds (widget information that you own), let's display that next time (for ease of understanding, specify in vscode to display the preceding and following lines when searching)

```
1 比較用 ・ 番号がそろってる状態. json:
2   126         ],
3   127:         "mediaIds": [
4   128           543
5
6   243         ],
7   244:         "mediaIds": [
8   245           543
9
10  283         ],
11  284:         "mediaIds": [
12  285           544
13
14  400         ],
15  401:         "mediaIds": [
16  402           543
17
18  440         ],
19  441:         "mediaIds": [
20  442           544
21
22  480         ],
23  481:         "mediaIds": [
24  482           545
25
26  597         ],
27  598:         "mediaIds": [
28  599           543
29
30  637         ],
```



31	638:	"mediaIds": [
32	639	544
33		
34	677	],
35	678:	"mediaIds": [
36	679	545
37		
38	717	],
39	718:	"mediaIds": [
40	719	546
41		
42	834	],
43	835:	"mediaIds": [
44	836	543
45		
46	874	],
47	875:	"mediaIds": [
48	876	544
49		
50	914	],
51	915:	"mediaIds": [
52	916	545
53		
54	954	],
55	955:	"mediaIds": [
56	956	546
57		
58	994	],
59	995:	"mediaIds": [
60	996	547
61		
62	1111	],
63	1112:	"mediaIds": [
64	1113	543
65		
66	1151	],
67	1152:	"mediaIds": [
68	1153	544
69		
70	1191	],
71	1192:	"mediaIds": [
72	1193	545
73		
74	1231	],
75	1232:	"mediaIds": [
76	1233	546
77		
78	1271	],
79	1272:	"mediaIds": [
80	1273	547
81		
82	1311	],
83	1312:	"mediaIds": [
84	1313	548

85  
86 1428 ],  
87 1429: "mediaIds": [  
88 1430 543  
89  
90 1468 ],  
91 1469: "mediaIds": [  
92 1470 544  
93  
94 1508 ],  
95 1509: "mediaIds": [  
96 1510 545  
97  
98 1548 ],  
99 1549: "mediaIds": [  
100 1550 546  
101  
102 1588 ],  
103 1589: "mediaIds": [  
104 1590 547  
105  
106 1628 ],  
107 1629: "mediaIds": [  
108 1630 548  
109  
110 1668 ],  
111 1669: "mediaIds": [  
112 1670 549  
113  
114 1785 ],  
115 1786: "mediaIds": [  
116 1787 543  
117  
118 1825 ],  
119 1826: "mediaIds": [  
120 1827 544  
121  
122 1865 ],  
123 1866: "mediaIds": [  
124 1867 545  
125  
126 1905 ],  
127 1906: "mediaIds": [  
128 1907 546  
129  
130 1945 ],  
131 1946: "mediaIds": [  
132 1947 547  
133  
134 1985 ],  
135 1986: "mediaIds": [  
136 1987 548  
137  
138 2025 ],

```

139 2026:      "mediaIds": [
140 2027          549
141
142 2065      ],
143 2066:      "mediaIds": [
144 2067          550
145
146 2182      ],
147 2183:      "mediaIds": [
148 2184          543
149
150 2222      ],
151 2223:      "mediaIds": [
152 2224          544
153
154 2262      ],
155 2263:      "mediaIds": [
156 2264          545
157
158 2302      ],
159 2303:      "mediaIds": [
160 2304          546
161
162 2342      ],
163 2343:      "mediaIds": [
164 2344          547
165
166 2382      ],
167 2383:      "mediaIds": [
168 2384          548
169
170 2422      ],
171 2423:      "mediaIds": [
172 2424          549
173
174 2462      ],
175 2463:      "mediaIds": [
176 2464          550
177
178 2502      ],
179 2503:      "mediaIds": [
180 2504          551

```

Comparing this with the widgetId in the query result below, we can confirm that media numbers 0 to 9 are arranged in the correct order.

画像名	widgetId	playlistId	ownerId	type	duration	displayOrder	calculatedDuration	useDuration	fromDt	toDt	createdDt	modifiedDt	mediaId	name
TestRenban_0.jpg	1169	343	10	image	15	1	15	1	0	2147483647	1694594883	1694594883	543	TestRenban_0.jpg
TestRenban_1.jpg	1170	343	10	image	15	2	15	1	0	2147483647	1694594883	1694594883	544	TestRenban_1.jpg
TestRenban_2.jpg	1171	343	10	image	15	3	15	1	0	2147483647	1694594883	1694594883	545	TestRenban_2.jpg
TestRenban_3.jpg	1172	343	10	image	15	4	15	1	0	2147483647	1694594883	1694594883	546	TestRenban_3.jpg
TestRenban_4.jpg	1173	343	10	image	15	5	15	1	0	2147483647	1694594883	1694594883	547	TestRenban_4.jpg
TestRenban_5.jpg	1174	343	10	image	15	6	15	1	0	2147483647	1694594883	1694594883	548	TestRenban_5.jpg
TestRenban_6.jpg	1175	343	10	image	15	7	15	1	0	2147483647	1694594883	1694594883	549	TestRenban_6.jpg
TestRenban_7.jpg	1176	343	10	image	15	8	15	1	0	2147483647	1694594883	1694594883	550	TestRenban_7.jpg
TestRenban_8.jpg	1177	343	10	image	15	9	15	1	0	2147483647	1694594883	1694594883	551	TestRenban_8.jpg
TestRenban_9.jpg	1178	343	10	image	15	10	15	1	0	2147483647	1694594883	1694594883	552	TestRenban_9.jpg

Now you can see how \$media information is sequentially added to the variable \$playlist.

```
244 */
245 private function createAndAssign($playlist, $media, $displayOrder)
246 {
247     $this->log->debug('Media Item needs to be assigned ' . $media->name . ' in sequence ' . $displayOrder);
248
249     //検証用
250     if($playlist->name == "No2_2023/09/01 16:17 cronタスクのテスト用 TestRenban_") {
251         error_log("No2_2023/09/01 16:17 cronタスクのテスト用 TestRenban_");
252         error_log(json_encode($playlist));
253         error_log(json_encode($media));
254         error_log("表示順序 : " . $displayOrder);
255     }
256 }
```

As for the order, the quickest way is to look at the filename of the variable \$media.

```
▼ { } 番号がそろってる状態.json 比較用
  "fileName": "TestRenban_0.jpg",
  "fileName": "TestRenban_1.jpg",
  "fileName": "TestRenban_2.jpg",
  "fileName": "TestRenban_3.jpg",
  "fileName": "TestRenban_4.jpg",
  "fileName": "TestRenban_5.jpg",
  "fileName": "TestRenban_6.jpg",
  "fileName": "TestRenban_7.jpg",
  "fileName": "TestRenban_8.jpg",
  "fileName": "TestRenban_9.jpg",
  ▼ { } 番号がそろってる状態.json 比較用
```

As for \$displayOrder, which is the most important display order, the results were shocking.

```
▼ { } 番号がそろってる状態.json 比較用
  ] 表示順序 : 2
  ] 表示順序 : 2
  ] 表示順序 : 2
  ] 表示順序 : 2
  ] 表示順序 : 2
  ] 表示順序 : 2
  ] 表示順序 : 2
  ] 表示順序 : 2
  ] 表示順序 : 2
  ] 表示順序 : 2
```

Actually, this 2 is a counter used in the process of looping through all dynamic playlists (a counter used to count the number of all dynamic playlists), but by mistake, it was directly used for DisplayOrder

. There is.

Therefore, the above 2 merely indicates that the playlist is being processed second in the loop of the entire playlist.

```
DynamicPlaylistSyncTask.php C:\...XTR  DynamicPlaylistSyncTask.php 比較用 9+ ×  検索: displayOr...
比較用 > DynamicPlaylistSyncTask.php > DynamicPlaylistSyncTask > run
216
217         if ($assignmentMade) {
218             // We've made an assignment change, so audit this change
219             // don't audit any downstream save operations
220             $playlist->save([
221                 'auditPlaylist' => true,
222                 'audit' => false
223             ]);
224         }
225     } else {
226         $this->log->debug('No differences detected');
227     }
228
229 } catch (XiboException $exception) {
230     $this->log->debug($exception->getTraceAsString());
231     $this->log->error('Problem with PlaylistId: ' . $playlist->getId() . ',
232     $this->appendRunMessage('Error with Playlist: ' . $playlist->name);
233 }
234 }
235
236 $this->appendRunMessage('Updated ' . $count . ' Playlists');
237 }
238
```

In other words, it is inappropriate to use \$count in this loop, and the problem is that the following loop does not have a unique counter.

```
error_log(json_encode($media));

// Do we have any mediaId's left which should be assigned and aren't?
// Add the ones we have left
foreach ($media as $item) {
    if (in_array($item->mediaId, $mediaIds)) {
        $assignmentMade = true;
        $this->createAndAssign($playlist, $item, $count);
    }
}
}
```

Before making any modifications, let's take a look at the following example, which is the issue in question.

This time, I threw in 0 to 9 without ClearAll (delete/recreate)

⇒ It was normal. This is probably because the entire playlist is now clean even without ClearAll because it is deleted and recreated.

Now, try deleting and recreating only 5 to 9.

⇒ It was normal. Perhaps this is because it just attaches to the end of the existing 0-9.

However, based on the conclusion so far, the conditions for triggering are met: the existing media is already linked

, and multiple files are deleted and recreated, so it could be a coincidence ⇒ I did it two additional times, but the context is normal.



```
11 161: "mediaIds": [  
12 162 "575"  
13  
14 215 ],  
15 216: "mediaIds": [  
16 217 "576"  
17  
18 270 ],  
19 271: "mediaIds": [  
20 272 "577"  
21  
22 402 ],  
23 403: "mediaIds": [  
24 404 "573"  
25  
26 457 ],  
27 458: "mediaIds": [  
28 459 "574"  
29  
30 512 ],  
31 513: "mediaIds": [  
32 514 "575"  
33  
34 567 ],  
35 568: "mediaIds": [  
36 569 "576"  
37  
38 622 ],  
39 623: "mediaIds": [  
40 624 "577"  
41  
42 677 ],  
43 678: "mediaIds": [  
44 679 578  
45  
46 794 ],  
47 795: "mediaIds": [  
48 796 "573"  
49  
50 849 ],  
51 850: "mediaIds": [  
52 851 "574"  
53  
54 904 ],  
55 905: "mediaIds": [  
56 906 "575"  
57  
58 959 ],  
59 960: "mediaIds": [  
60 961 "576"  
61  
62 1014 ],  
63 1015: "mediaIds": [  
64 1016 "577"
```

65  
66 1069 ],  
67 1070: "mediaIds": [  
68 1071 578  
69  
70 1109 ],  
71 1110: "mediaIds": [  
72 1111 579  
73  
74 1226 ],  
75 1227: "mediaIds": [  
76 1228 "573"  
77  
78 1281 ],  
79 1282: "mediaIds": [  
80 1283 "574"  
81  
82 1336 ],  
83 1337: "mediaIds": [  
84 1338 "575"  
85  
86 1391 ],  
87 1392: "mediaIds": [  
88 1393 "576"  
89  
90 1446 ],  
91 1447: "mediaIds": [  
92 1448 "577"  
93  
94 1501 ],  
95 1502: "mediaIds": [  
96 1503 578  
97  
98 1541 ],  
99 1542: "mediaIds": [  
100 1543 579  
101  
102 1581 ],  
103 1582: "mediaIds": [  
104 1583 580  
105  
106 1698 ],  
107 1699: "mediaIds": [  
108 1700 "573"  
109  
110 1753 ],  
111 1754: "mediaIds": [  
112 1755 "574"  
113  
114 1808 ],  
115 1809: "mediaIds": [  
116 1810 "575"  
117  
118 1863 ],



```

119 1864:      "mediaIds": [
120 1865          "576"
121
122 1918      ],
123 1919:      "mediaIds": [
124 1920          "577"
125
126 1973      ],
127 1974:      "mediaIds": [
128 1975          578
129
130 2013      ],
131 2014:      "mediaIds": [
132 2015          579
133
134 2053      ],
135 2054:      "mediaIds": [
136 2055          580
137
138 2093      ],
139 2094:      "mediaIds": [
140 2095          581

```

Since I can see some regularity, I demonstrated the movement with gpt4.

The implementation of assignWidget called by SyncDynamicPlaylist is as follows

<https://github.com/xibosignage/xibo-cms/blob/369c6570d8fa28db1907a6302e5a0069781eb551/lib/Entity/Playlist.php>

```

1 以下の情報0、1、2から、displayOrderの決まる規則性を見つけ出せ
2
3 情報0
4 情報1のうち以下の5つは、削除・再作成を通じた更新分であり、その過程を示したのが情報2
5 TestRenban_0. jpg
6 TestRenban_1. jpg
7 TestRenban_2. jpg
8 TestRenban_3. jpg
9 TestRenban_4. jpg
10
11 情報1
12 # displayOrder, mediaId, name
13 '1', '573', 'TestRenban_5. jpg'
14 '2', '578', 'TestRenban_0. jpg'
15 '3', '574', 'TestRenban_6. jpg'
16 '4', '579', 'TestRenban_1. jpg'
17 '5', '575', 'TestRenban_7. jpg'
18 '6', '580', 'TestRenban_2. jpg'
19 '7', '576', 'TestRenban_8. jpg'
20 '8', '581', 'TestRenban_3. jpg'
21 '9', '577', 'TestRenban_9. jpg'
22 '10', '582', 'TestRenban_4. jpg'
23
24 情報2
25 mediaId: 573
26 mediaId: 574

```

27 mediaId: 575  
28 mediaId: 576  
29 mediaId: 577  
30 —  
31 mediaId: 573  
32 mediaId: 574  
33 mediaId: 575  
34 mediaId: 576  
35 mediaId: 577  
36 mediaId: 578  
37 —  
38 mediaId: 573  
39 mediaId: 574  
40 mediaId: 575  
41 mediaId: 576  
42 mediaId: 577  
43 mediaId: 578  
44 mediaId: 579  
45 —  
46 mediaId: 573  
47 mediaId: 574  
48 mediaId: 575  
49 mediaId: 576  
50 mediaId: 577  
51 mediaId: 578  
52 mediaId: 579  
53 mediaId: 580  
54 —  
55 mediaId: 573  
56 mediaId: 574  
57 mediaId: 575  
58 mediaId: 576  
59 mediaId: 577  
60 mediaId: 578  
61 mediaId: 579  
62 mediaId: 580  
63 mediaId: 581

64 ChatGPT

65 情報を解析すると、以下の規則性が見えます：

66

67 情報0では5つのファイルが削除・再作成されることが明示されています。これらのファイルは TestRenban\_0.jpg から TestRe

68

69 情報1は最終的な表示順序(displayOrder)、メディアID(mediaId)、そして名前(name)を示しています。ここで名前が TestRenk

70

71 TestRenban\_0.jpg はメディアID '578' と表示順序 '2' を持つ

72 TestRenban\_1.jpg はメディアID '579' と表示順序 '4' を持つ

73 TestRenban\_2.jpg はメディアID '580' と表示順序 '6' を持つ

74 TestRenban\_3.jpg はメディアID '581' と表示順序 '8' を持つ

75 TestRenban\_4.jpg はメディアID '582' と表示順序 '10' を持つ

76 情報2はメディアの削除と再作成のプロセスを示しているようです。最初はID 573から577までのメディアが存在し、次に578が

77

78 したがって、displayOrderの決まる規則性は、新しいメディアが追加されるたびに、その名前と関連する新しいメディアIDが割

79

80

81

82

83 User

84 なるほど。では、以下のようなプロセスを経て情報1の状態になる過程をデモしてくれますか？

85

86 TestRenban\_0. jpg はメディアID '578' と表示順序 '2' を持つ

87 TestRenban\_1. jpg はメディアID '579' と表示順序 '4' を持つ

88 TestRenban\_2. jpg はメディアID '580' と表示順序 '6' を持つ

89 TestRenban\_3. jpg はメディアID '581' と表示順序 '8' を持つ

90 TestRenban\_4. jpg はメディアID '582' と表示順序 '10' を持つ

91 ChatGPT

92 はい、以下のプロセスで情報1の状態に至るデモを行います：

93

94 初期状態

95 メディアリスト

96 573 - TestRenban\_5. jpg

97 574 - TestRenban\_6. jpg

98 575 - TestRenban\_7. jpg

99 576 - TestRenban\_8. jpg

100 577 - TestRenban\_9. jpg

101 578 - TestRenban\_0. jpg (まだ存在しない)

102 579 - TestRenban\_1. jpg (まだ存在しない)

103 580 - TestRenban\_2. jpg (まだ存在しない)

104 581 - TestRenban\_3. jpg (まだ存在しない)

105 582 - TestRenban\_4. jpg (まだ存在しない)

106 ステップ1: TestRenban\_0. jpg の追加

107 573 - TestRenban\_5. jpg

108 578 - TestRenban\_0. jpg (新規追加)

109 574 - TestRenban\_6. jpg

110 575 - TestRenban\_7. jpg

111 576 - TestRenban\_8. jpg

112 577 - TestRenban\_9. jpg

113 ステップ2: TestRenban\_1. jpg の追加

114 573 - TestRenban\_5. jpg

115 578 - TestRenban\_0. jpg

116 574 - TestRenban\_6. jpg

117 579 - TestRenban\_1. jpg (新規追加)

118 575 - TestRenban\_7. jpg

119 576 - TestRenban\_8. jpg

120 577 - TestRenban\_9. jpg

121 ステップ3: TestRenban\_2. jpg の追加

122 573 - TestRenban\_5. jpg

123 578 - TestRenban\_0. jpg

124 574 - TestRenban\_6. jpg

125 579 - TestRenban\_1. jpg

126 575 - TestRenban\_7. jpg

127 580 - TestRenban\_2. jpg (新規追加)

128 576 - TestRenban\_8. jpg

129 577 - TestRenban\_9. jpg

130 ステップ4: TestRenban\_3. jpg の追加

131 573 - TestRenban\_5. jpg

132 578 - TestRenban\_0. jpg

133 574 - TestRenban\_6. jpg

134 579 - TestRenban\_1. jpg

135 575 - TestRenban\_7. jpg  
136 580 - TestRenban\_2. jpg  
137 576 - TestRenban\_8. jpg  
138 581 - TestRenban\_3. jpg (新規追加)  
139 577 - TestRenban\_9. jpg  
140 ステップ5: TestRenban\_4. jpg の追加  
141 573 - TestRenban\_5. jpg  
142 578 - TestRenban\_0. jpg  
143 574 - TestRenban\_6. jpg  
144 579 - TestRenban\_1. jpg  
145 575 - TestRenban\_7. jpg  
146 580 - TestRenban\_2. jpg  
147 576 - TestRenban\_8. jpg  
148 581 - TestRenban\_3. jpg  
149 577 - TestRenban\_9. jpg  
150 582 - TestRenban\_4. jpg (新規追加)  
151 このプロセスを経て、情報 1 の状態になります。さらに、各ステップでのメディアIDの変更が情報 2 のデータに一致します。  
152  
153  
154  
155  
156 User  
157  
158 7 / 7  
159  
160 ・新規追加はみなdisplayOrderの値が元々は2として挿入され、挿入後に現在のDisplayOrderにおちつきます。  
161 ・573～577は、新規追加が来る前はdisplayorderが6～10です。  
162 ・つまり新規追加分の578～582は、新規追加分として来る前はdisplayorderが1～5です。  
163 ・ソートロジックは以下のPHPです  
164 ・PHP  
165 ヒント: いかは  
166 /\*\*  
167 \* @param Widget \$widget  
168 \* @param int \$displayOrder  
169 \*/  
170 public function assignWidget(\$widget, \$displayOrder = null)  
171 {  
172 \$this->load();  
173  
174 // Has a display order been provided?  
175 if (\$displayOrder !== null) {  
176 // We need to shuffle any existing widget down to make space for this one.  
177 foreach (\$this->widgets as \$existingWidget) {  
178 if (\$existingWidget->displayOrder < \$displayOrder) {  
179 // Earlier in the list, so do nothing.  
180 continue;  
181 } else {  
182 // This widget is >= the display order and therefore needs to be moved down one position.  
183 \$existingWidget->displayOrder = \$existingWidget->displayOrder + 1;  
184 }  
185  
186 // Set the incoming widget to the requested display order.  
187 \$widget->displayOrder = \$displayOrder;  
188 }  
189 }  
190 }

```
189     } else {
190         // Take the next available one
191         $widget->displayOrder = count($this->widgets) + 1;
192     }
193
194     $this->widgets[] = $widget;
195 }
```

196 ・最終的には以下の順序になります。

197 573 - TestRenban\_5. jpg

198 578 - TestRenban\_0. jpg

199 574 - TestRenban\_6. jpg

200 579 - TestRenban\_1. jpg

201 575 - TestRenban\_7. jpg

202 580 - TestRenban\_2. jpg

203 576 - TestRenban\_8. jpg

204 581 - TestRenban\_3. jpg

205 577 - TestRenban\_9. jpg

206 582 - TestRenban\_4. jpg

207 ここから見える規則性をデモで示してください

208 ChatGPT

209 了解しました。PHPコードと提供された情報を基に、デモを行います。コードの動作に基づいて、新規追加分（578～582）が順

210

211 初期状態

212 メディアリスト

213 573 - TestRenban\_5. jpg

214 574 - TestRenban\_6. jpg

215 575 - TestRenban\_7. jpg

216 576 - TestRenban\_8. jpg

217 577 - TestRenban\_9. jpg

218 ステップ1: TestRenban\_0. jpg (578) の追加

219 指定の displayOrder は 2

220

221 573のdisplayOrderは2より小さいため何もしない。

222 574のdisplayOrderは2以上なので1つ増やして3になる。

223 575のdisplayOrderは3以上なので1つ増やして4になる。

224 576のdisplayOrderは4以上なので1つ増やして5になる。

225 577のdisplayOrderは5以上なので1つ増やして6になる。

226 メディアリスト

227 573 - TestRenban\_5. jpg

228 578 - TestRenban\_0. jpg (新規追加)

229 574 - TestRenban\_6. jpg

230 575 - TestRenban\_7. jpg

231 576 - TestRenban\_8. jpg

232 577 - TestRenban\_9. jpg

233 ステップ2: TestRenban\_1. jpg (579) の追加

234 指定の displayOrder は 4

235

236 573と578のdisplayOrderは4より小さいため何もしない。

237 574のdisplayOrderは4以上なので1つ増やして4になる。

238 575のdisplayOrderは4以上なので1つ増やして5になる。

239 576のdisplayOrderは5以上なので1つ増やして6になる。

240 577のdisplayOrderは6以上なので1つ増やして7になる。

241 メディアリスト

242 573 - TestRenban\_5. jpg

243 578 - TestRenban\_0. jpg  
244 574 - TestRenban\_6. jpg  
245 579 - TestRenban\_1. jpg (新規追加)  
246 575 - TestRenban\_7. jpg  
247 576 - TestRenban\_8. jpg  
248 577 - TestRenban\_9. jpg  
249 ステップ3: TestRenban\_2. jpg (580) の追加  
250 指定の displayOrder は 6  
251  
252 573, 578, 574, 579のdisplayOrderは6より小さいため何もしない。  
253 575のdisplayOrderは6以上なので1つ増やして6になる。  
254 576のdisplayOrderは6以上なので1つ増やして7になる。  
255 577のdisplayOrderは7以上なので1つ増やして8になる。  
256 メディアリスト  
257 573 - TestRenban\_5. jpg  
258 578 - TestRenban\_0. jpg  
259 574 - TestRenban\_6. jpg  
260 579 - TestRenban\_1. jpg  
261 575 - TestRenban\_7. jpg  
262 580 - TestRenban\_2. jpg (新規追加)  
263 576 - TestRenban\_8. jpg  
264 577 - TestRenban\_9. jpg  
265 ステップ4: TestRenban\_3. jpg (581) の追加  
266 指定の displayOrder は 8  
267  
268 573, 578, 574, 579, 575, 580のdisplayOrderは8より小さいため何もしない。  
269 576のdisplayOrderは8以上なので1つ増やして8になる。  
270 577のdisplayOrderは8以上なので1つ増やして9になる。  
271 メディアリスト  
272 573 - TestRenban\_5. jpg  
273 578 - TestRenban\_0. jpg  
274 574 - TestRenban\_6. jpg  
275 579 - TestRenban\_1. jpg  
276 575 - TestRenban\_7. jpg  
277 580 - TestRenban\_2. jpg  
278 576 - TestRenban\_8. jpg  
279 581 - TestRenban\_3. jpg (新規追加)  
280 577 - TestRenban\_9. jpg  
281 ステップ5: TestRenban\_4. jpg (582) の追加  
282 指定の displayOrder は 10, 新規追加なのでリストの最後に追加され、displayOrder は 10  
283  
284 メディアリスト  
285 573 - TestRenban\_5. jpg  
286 578 - TestRenban\_0. jpg  
287 574 - TestRenban\_6. jpg  
288 579 - TestRenban\_1. jpg  
289 575 - TestRenban\_7. jpg  
290 580 - TestRenban\_2. jpg  
291 576 - TestRenban\_8. jpg  
292 581 - TestRenban\_3. jpg  
293 577 - TestRenban\_9. jpg  
294 582 - TestRenban\_4. jpg (新規追加)  
295 以上のプロセスによって、指定された最終順序が達成されました。このデモは、指定されたPHPコードのassignWidget関数を使  
296

```
297
298
299
300 User
301 assignWidgetのソートの概要を簡潔に示してくれますか？
302 ChatGPT
303 assignWidget関数は、新しいウィジェットを特定の位置に挿入し、その位置から後ろのすべてのウィジェットの表示順序を1つ
304
305 ウィジェットのロード：最初に現在のウィジェットリストをロードします。
306
307 displayOrderの確認：displayOrderが指定されている場合（nullでない）、以下のステップが実行されます：
308
309 既存のウィジェットをループして特定の位置に新しいウィジェットを挿入します。
310 指定されたdisplayOrder以上のdisplayOrderを持つすべてのウィジェットは、そのdisplayOrderが1つずつ増加します（シフト）
311 新しいウィジェットは指定されたdisplayOrderに設定されます。
312 新しいdisplayOrderの割り当て：displayOrderが指定されていない場合（nullの場合）、新しいウィジェットはリストの最後に
313
314 ウィジェットの追加：新しいウィジェットはウィジェットリストの最後に追加されます。
315
316 この関数によって、ウィジェットは必要に応じて適切な位置に追加や再配置され、ウィジェットリストは整然と保たれます。
```

This is the reason why even 10 files with the same displayOrder are arranged in an orderly manner.

---

Finally, the following points were corrected. "If media is deleted or recreated even for one element in a sub-playlist, the entire widget of the sub-playlist will be

---