

Example

Create Workspaces

by:	Alex del Bondio
published:	August 2018
description:	Macros to create a custom set of workspaces.
remarks:	Used with a new show as you cannot import workspaces yet record , workspace , delete , windows

The code below only is a snippet of the full macro. The full macro can be found [here](#)

adb_workspace_creation.xml

and the content additional to the snippet is laid out further down.

functions

- [UserMacros.RecallMacroById](#)
- [Windows.Scrolling.Vertical.Scroll](#)
- [Workspace.Record](#)
- [Handles.SetSourceHandle](#)
- [Handles.SetUserNumber](#)
- [Handles.SetSourceHandleFromHandle](#)
- [Handles.ConfirmDelete](#)
- [Handles.SetSourceHandleRange](#)
- [Windows.SetWindowProperty.X](#)
- [Windows.SetWindowProperty.Y](#)
- [Windows.SetWindowProperty.Width](#)
- [Windows.SetWindowProperty.Height](#)

affected properties

- the respective window's attributes (x, y, height, width), see [Windows](#)
- [Windows.windowId.ButtonSize](#)
- [Windows.windowId.TextSize](#)
- [Windows.windowId.FixedColumns](#)
- [Windows.windowId.FixedRows](#)
- [Windows.windowId.Pages](#)
- [Windows.windowId.CurrentPage](#)
- [Handles.AllowEditWorkspaces](#)

See also [Windows](#) for more details about workspace windows.

control structures

- [menuLink](#)

Code

[adb_workspace_creation_snippet.xml](#)

```
<?xml version="1.0" encoding="utf-8"?>
<avolites.macros>

<!-- macros to run macros as specified -->
  <macro id="adb.Macros.createworkspaces" name="adb create workspaces">
    <description>Runs the following macros as specified.</description>
    <sequence>
      <step
pause="0.001">UserMacros.RecallMacroById("adb.Macros.deleteWorkspace1thru24")</step>

      <!--01-->
      <step
pause="0.001">UserMacros.RecallMacroById("adb.Macros.CloseAllWindows")</step>
      <step
pause="0.001">UserMacros.RecallMacroById("adb.Macros.ws.Playbacks")</step>
      <step
pause="0.001">UserMacros.RecallMacroById("adb.Macros.recordworkspace1")</step>

      <!--02-->
      <step
pause="0.001">UserMacros.RecallMacroById("adb.Macros.CloseAllWindows")</step>
      <step
pause="0.001">UserMacros.RecallMacroById("adb.Macros.ws.Program")</step>
      <step
pause="0.001">UserMacros.RecallMacroById("adb.Macros.recordworkspace2")</step>

      <!-- setup and record some more workspaces -->

    </sequence>
  </macro>

<!-- Macros to Open Predefined Workspace Windows -->
<!-- Windows.SetWindowRegion("Windows.ChannelGrid", "secondary") needs to be implemented -->
<!-- maximised="False" regionId="primary" scrollOffset seemingly not functional -->

  <macro id="adb.Macros.ws.Playbacks" name="adb ws 01_Playbacks">
```

```
<description>opens Workspace Playbacks</description>
<sequence>
  <step pause="0.001">
    <menuLink id="Windows.Playbacks" stack="mainWindowStack"
behaviour="PushOrRaise" maximised="True" regionId="primary" />
  </step>
  <step
pause="0.001">UserMacros.RecallMacroById("adb.Macros.MinimiseWindow")</
step>
  <step>Windows.SetWindowProperty.X("Windows.Playbacks", 0)</step>
  <step>Windows.SetWindowProperty.Y("Windows.Playbacks", 0)</step>
  <step>Windows.SetWindowProperty.Width("Windows.Playbacks",
2)</step>
  <step>Windows.SetWindowProperty.Height("Windows.Playbacks",
2)</step>
<step>ActionScript.SetProperty.Enum("Windows.Playbacks.ButtonSize",
"Fixed")</step>
  <step>ActionScript.SetProperty.Enum("Windows.Playbacks.TextSize",
"Global")</step>
<step>ActionScript.SetProperty.Integer("Windows.Playbacks.FixedColumns"
, 12)</step>
<step>ActionScript.SetProperty.Integer("Windows.Playbacks.FixedRows",
12)</step>
  <step>ActionScript.SetProperty.Boolean("Windows.Playbacks.Pages",
true)</step>
<step>ActionScript.SetProperty.Integer("Windows.Playbacks.CurrentPage",
0)</step>
  <step>Windows.Scrolling.Vertical.Scroll(-100000)</step>
</sequence>
</macro>
<macro id="adb.Macros.ws.Program" name="adb ws 02_Program">
  <description>opens Workspace Program</description>
  <sequence>
    <step pause="0.001">
      <menuLink id="Windows.Groups" stack="mainWindowStack"
behaviour="PushOrRaise" maximised="True" regionId="secondary" />
    </step>
    <step>Windows.SetWindowProperty.X("Windows.Groups", 0)</step>
    <step>Windows.SetWindowProperty.Y("Windows.Groups", 0)</step>
    <step>Windows.SetWindowProperty.Width("Windows.Groups", 1)</step>
    <step>Windows.SetWindowProperty.Height("Windows.Groups",
1)</step>
    <step pause="0.001">
      <menuLink id="Windows.Colours" stack="mainWindowStack"
behaviour="PushOrRaise" maximised="True" regionId="secondary" />
    </step>
    <step>Windows.SetWindowProperty.X("Windows.Colours", 1)</step>
    <step>Windows.SetWindowProperty.Y("Windows.Colours", 0)</step>
    <step>Windows.SetWindowProperty.Width("Windows.Colours",
1)</step>
    <step>Windows.SetWindowProperty.Height("Windows.Colours",
```

```
1)/</step>
  <step pause="0.001">
    <menuLink id="Windows.Positions" stack="mainWindowStack"
behaviour="PushOrRaise" maximised="True" regionId="secondary" />
  </step>
  <step>Windows.SetWindowProperty.X("Windows.Positions", 0)</step>
  <step>Windows.SetWindowProperty.Y("Windows.Positions", 1)</step>
  <step>Windows.SetWindowProperty.Width("Windows.Positions",
1)/</step>
  <step>Windows.SetWindowProperty.Height("Windows.Positions",
1)/</step>
  <step pause="0.001">
    <menuLink id="Windows.Beams" stack="mainWindowStack"
behaviour="PushOrRaise" maximised="True" regionId="secondary" />
  </step>
  <step>Windows.SetWindowProperty.X("Windows.Beams", 1)</step>
  <step>Windows.SetWindowProperty.Y("Windows.Beams", 1)</step>
  <step>Windows.SetWindowProperty.Width("Windows.Beams", 1)</step>
  <step>Windows.SetWindowProperty.Height("Windows.Beams", 1)</step>
</sequence>
</macro>

<!-- Macros to Record Workspace Windows -->
  <macro id="adb.Macros.recordworkspace1" name="adb record workspace1">
    <description>Record Workspace 1.</description>
    <sequence>
<step>Workspace.Record(Handles.CreateHandleReference("Workspaces", 0,
0), "Playbacks", 0, true, false)</step>
      <step>ActionScript.SetProperty("Handles.AllowEditWorkspaces",
true)</step>
      <step>Handles.SetSourceHandle("Workspaces", 0)</step>
      <step>ActionScript.SetProperty("Handles.CurrentUserNumber",
userNumber:1)</step>
      <step>Handles.SetUserNumber()</step>
      <step>Handles.ClearSelection()</step>
      <step>ActionScript.SetProperty("Handles.AllowEditWorkspaces",
false)</step>
    </sequence>
  </macro>
  <macro id="adb.Macros.recordworkspace2" name="adb record workspace2">
    <description>Record Workspace 2.</description>
    <sequence>
<step>Workspace.Record(Handles.CreateHandleReference("Workspaces", 0,
1), "Program", 0, false, true)</step>
      <step>ActionScript.SetProperty("Handles.AllowEditWorkspaces",
true)</step>
      <step>Handles.SetSourceHandle("Workspaces", 1)</step>
      <step>ActionScript.SetProperty("Handles.CurrentUserNumber",
userNumber:2)</step>
      <step>Handles.SetUserNumber()</step>
```

```

        <step>Handles.ClearSelection()</step>
        <step>ActionScript.SetProperty("Handles.AllowEditWorkspaces",
false)</step>
    </sequence>
</macro>

<!-- Macros to delete Workspace Windows -->
    <macro id="adb.Macros.deleteWorkspacel1thru24" name="adb delete
workspace 1 through 24">
        <description>adb Delete Workspace 1-24</description>
        <sequence>
            <step
pause="0.001">ActionScript.SetProperty.Boolean("Handles.AllowEditWorksp
aces", true)</step>
            <step
pause="0.001">Handles.SetSourceHandleFromHandle("workspaceHandleUN=1")<
/step>
            <step pause="0.001">Handles.ConfirmDelete()</step>
            <step
pause="0.001">Handles.SetSourceHandleFromHandle("workspaceHandleUN=2")<
/step>
            <step pause="0.001">Handles.ConfirmDelete()</step>
            <step pause="0.001">Handles.SetSourceHandleRange("Workspaces",
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11})</step>
            <step pause="0.001">Handles.ConfirmDelete()</step>
            <step
pause="0.001">ActionScript.SetProperty.Boolean("Handles.AllowEditWorksp
aces", false)</step>
        </sequence>
    </macro>

</avolites.macros>

```

Explanation

This explains the functional steps within the sequence. For all the other XML details please refer to [Formats and syntax](#)

The code above is only a snippet to give you an idea of what is going on in the macro, the full macro will include the following:

- This is a collection of macros that will delete existing workspaces and replace them with workspaces defined in this code
- This collection is divided into labeled subparts. Best practice is to collapse all lines and go through it step by step
- The main macro adb create workspaces will only run other macros without further code
- macros to run macros as specified is the main macro that will run macros as defined
- Macros to Open Predefined Workspace Windows these macros will open predefined

window configurations

- **Macros to Open other Windows** are macros to open workspace windows as you know from the console. Included are windows you'll only find in certain menus (e.g. triggers)
- **Macros to Record Workspace Windows** these macros will record the actual workspaces
- **Macros to delete Workspace Windows** will delete all existing workspaces on the handles and unassigned ones with user number 1-24
- **Macros to Control Workspace Windows** are stock macros to control window size, position etc
- **Show Library options** Show Library display options which I use for my show (one column of macros right next to the show library window)
- **Channel Grid options** Channel Grid display options - same as for Show Library
- **Playback View / Cue View options** stock macros to control playback/cuelist view (e.g. next/prev cue)

How to use it

1. [make this macro available](#)
2. This macro is intended to be called when a new show is started - it deletes the factory workspaces and sets up new workspaces.

From:

<https://avosupport.de/wiki/> - **AVOSUPPORT**

Permanent link:

<https://avosupport.de/wiki/macros/example/createworkspaces?rev=1535283225>

Last update: **2018/08/26 11:33**

