

CN_TouchScreen_APROL R4.2 如何实现触摸屏画面全屏的方法

Exported from Confluence on 2024 January 26

Table of Contents

- 触摸屏全屏显示需求 3
- 通用的全屏实现的设置 3
- 全屏实现后的相关问题和解决..... 4
- 另一种可行的全屏替代方案 5
- 方案选择 10
- 后续—通用方案的补充更新 11

· 触摸屏全屏显示需求

需求

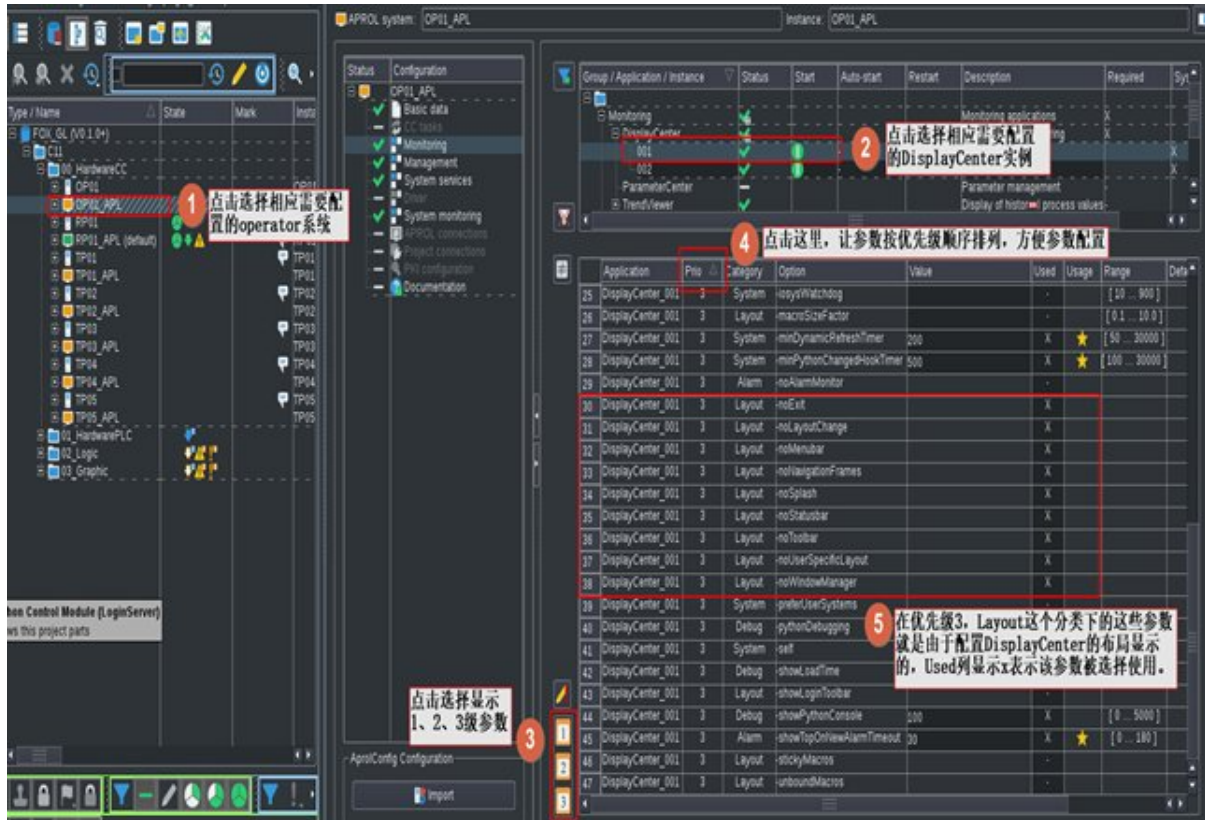
APROL 项目在画面呈现内容繁多时需要更大的显示空间，或着在对现场操作工操作规范比较严格的情况下，客户都希望最终操作的界面最好是全屏的，不需要其它额外的和操作无关的程序，且希望保持操作运行界面一直显示不能随意关闭退出，对于这种需求，我们可以通过对工程项目里相应的 **DisplayCenter** 的参数做相关的配置加以实现。如下图说明所示，要把这个 **title bar** 给隐藏掉，这样 **DisplayCenter** 也就实现了一直显示无法随意关闭；同时这样也非常方便客户去定制化自己风格的画面。



· 通用的全屏实现的设置

通用方案

DisplayCenter 画面需要全屏显示并且没有最顶上的 **title bar**，那么需要如下图去配置对相应的 **DisplayCenter** 的参数，主要是 **Layout** 这类参数，根据具体的需求相应去选择是否配置使用这些参数，至于每项参数的意义请具体参看帮助说明。而要实现 **DisplayCenter** 打开后没有最上面的 **title bar**，那么就需要选择参数项 **-noWindowManager**，这个参数项选择使用后，那么最终相应的 **DisplayCenter** 打开后就不会有 **title bar**。



· 全屏实现后的相关问题和解决

通用方案问题以及解决

如上面描述的通过对 **DisplayCenter** 的参数进行特定的设置后，可以实现 **DisplayCenter** 的全屏。这种方式实现的全屏，在实际运用中遇到过下面的几种情况，并给出了相应的解决方法供参考。

1. DisplayCenter 在使用 -noWindowManager 时的问题

在如上面对 **DisplayCenter** 配置 **-noWindowManager** 实现 **DisplayCenter** 全屏后，在某些情况下仍会导致 **title bar** 又会出现的情况。目前已知有如下情形：

- 如在 **DisplayCenter** 程序打开之前有带有 **title bar** 的第三方弹窗出现，如 **DisplayCenter** 设置为自动启动，但是该操作员站没有相应的 **TG** 授权，那么系统会弹出相应的 **APROL licensing** 的相应警告信息，点击该警告信息的“ok”按钮后，**DisplayCenter** 打开后就会显示 **title bar**。
- DisplayCenter** 打开并且没有 **title bar** 的情况下，如果页面调用了 **python** 编写的 **login/logout** 的按钮用于操作用户的切换，那么点击这个按钮后，会弹出用户 **LoginServer** 的输入用户和密码的弹窗，该弹窗是带有 **title bar** 的，那么完成这个用户和密码的输入确认后，**DisplayCenter** 马上就变成带 **title bar** 了。即 **DisplayCenter** 使用过程中，只要有带 **title bar** 的相应弹窗出现，确认后就会导致 **DisplayCenter** 出现 **title bar**。

因此，需要在项目中预先考虑上面的这些情况，禁止这些情形的出现。如果这种情况出现，临时的隐藏 **DisplayCenter** 的 **title bar** 的解决方法如下图三说明，把光标放置在 **title bar** 的区域里，点击鼠标右键，然后在出现的弹窗，选择 **More Actions --> No Border** 后，上面的 **title bar** 会消失。但是如果这个 **DisplayCenter** 关掉后重新打开，那么打开的 **DisplayCenter** 还是带 **title bar** 的，需要按上面的步骤去重新设置一次。



2. 全屏下的虚拟键盘调用问题

如果 **DisplayCenter** 设置 **-noWindowManager** 参数，那么相应操作员站的 **DisplayCenter** 打开后是没有 **title bar** 的。在这使用触摸屏的情况时，如果相应的画面里有需要输入，则需要调用虚拟键盘 **xvkbd**，有关触摸屏下如何调用和定制化虚拟键盘，请参看文档《**APROL R4.2 使用触摸屏时如何调用和定制虚拟键盘**》。

在 **DisplayCenter** 没有 **title bar** 情况下，我们发现在点击画面上的输入区域调用虚拟键盘时，虚拟键盘的应用界面无法弹出在 **DisplayCenter** 画面之前被加以使用，相当于虚拟键盘没有调用，但实际经常在后台已经打开，只是因为 **DisplayCenter** 设置了 **-noWindowManager** 导致虚拟键盘无法自动弹出到 **DisplayCenter** 之上。

要解决虚拟键盘无法自动弹出的问题，临时的一个解决方法就是 **DisplayCenter** 里面不要去设置 **-noWindowManager** 参数，在 **DisplayCenter** 有 **title bar** 的情况下，点击画面上的输入区域时虚拟键盘可以自动弹出，没有问题，但是前提是 **DisplayCenter** 里面不要去设置 **-noWindowManager** 参数。

· 另一种可行的全屏替代方案

替代方案

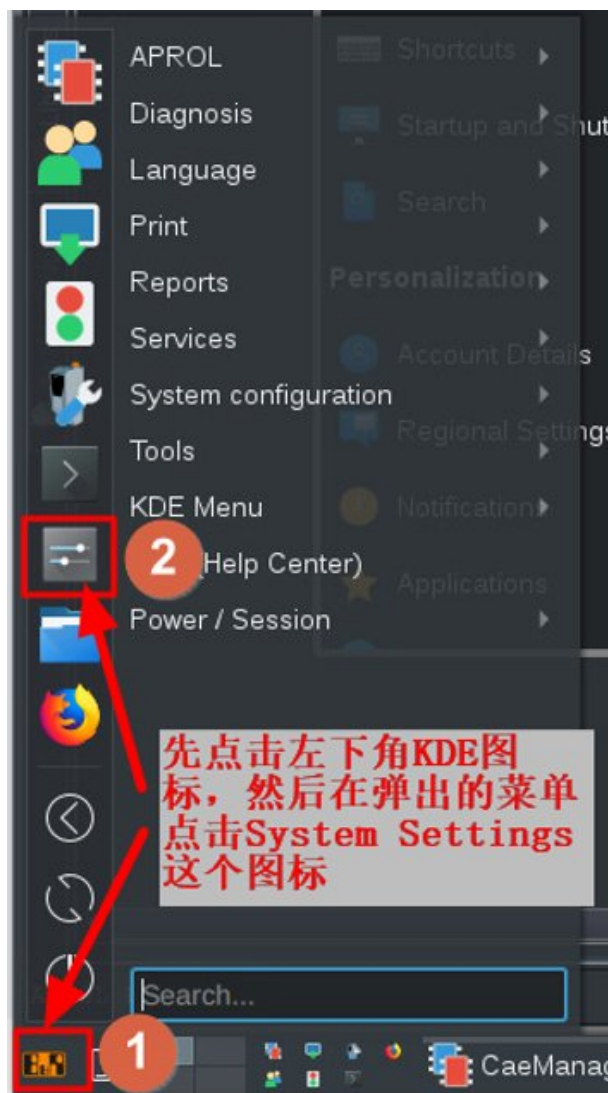
从上面的描述上来看，如果要想实现 **DisplayCenter** 全屏没有 **title bar**，同时调用虚拟键盘又没有问题的话，相当于是不可行的。

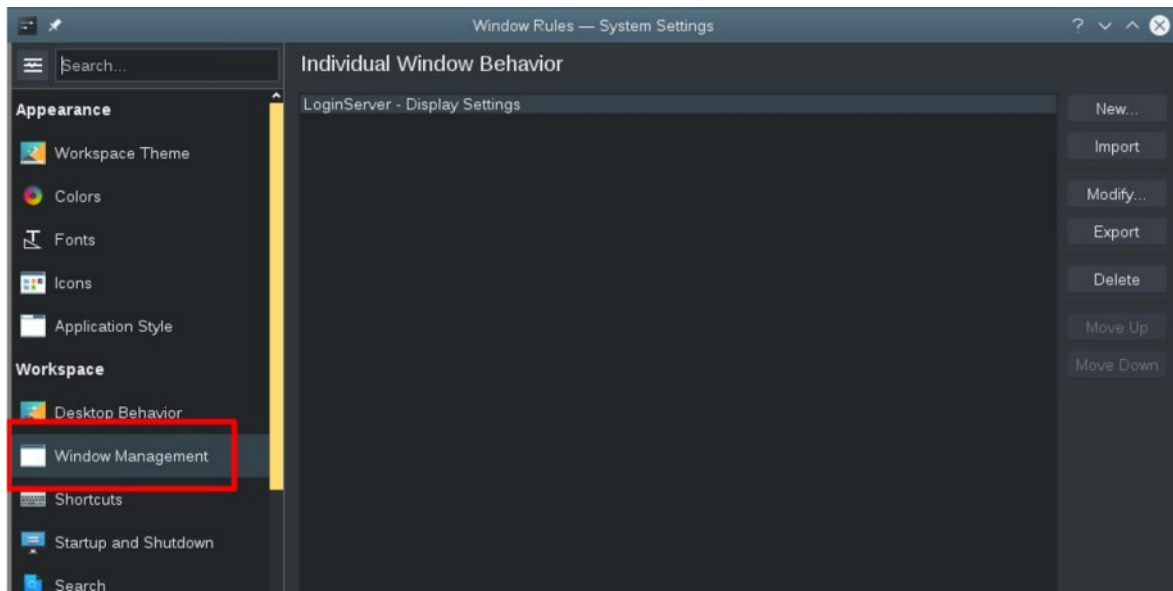
在经过多方尝试的基础上，测试后得出如下的一个替代方案。

首先，项目里面对 **DisplayCenter** 的 **-noWindowManager** 这个参数不要去勾选使用，接下来，我们需要在相应的操作员站上做下面的两步工作。

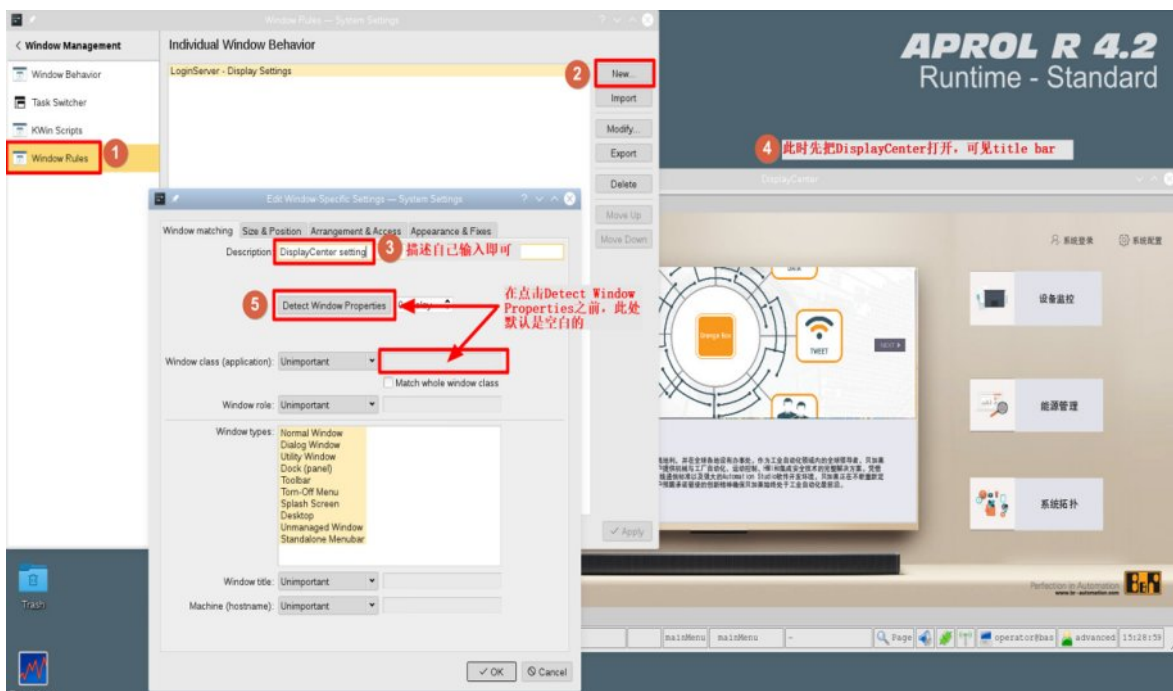
1. 创建 DisplayCenter 的 Window Management 的规则

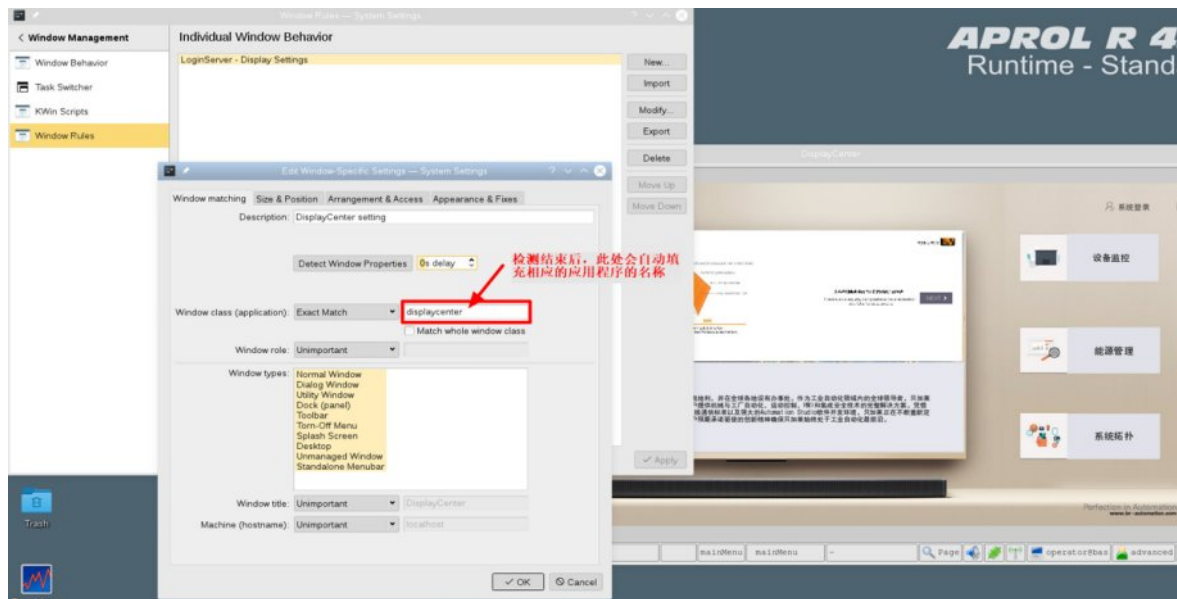
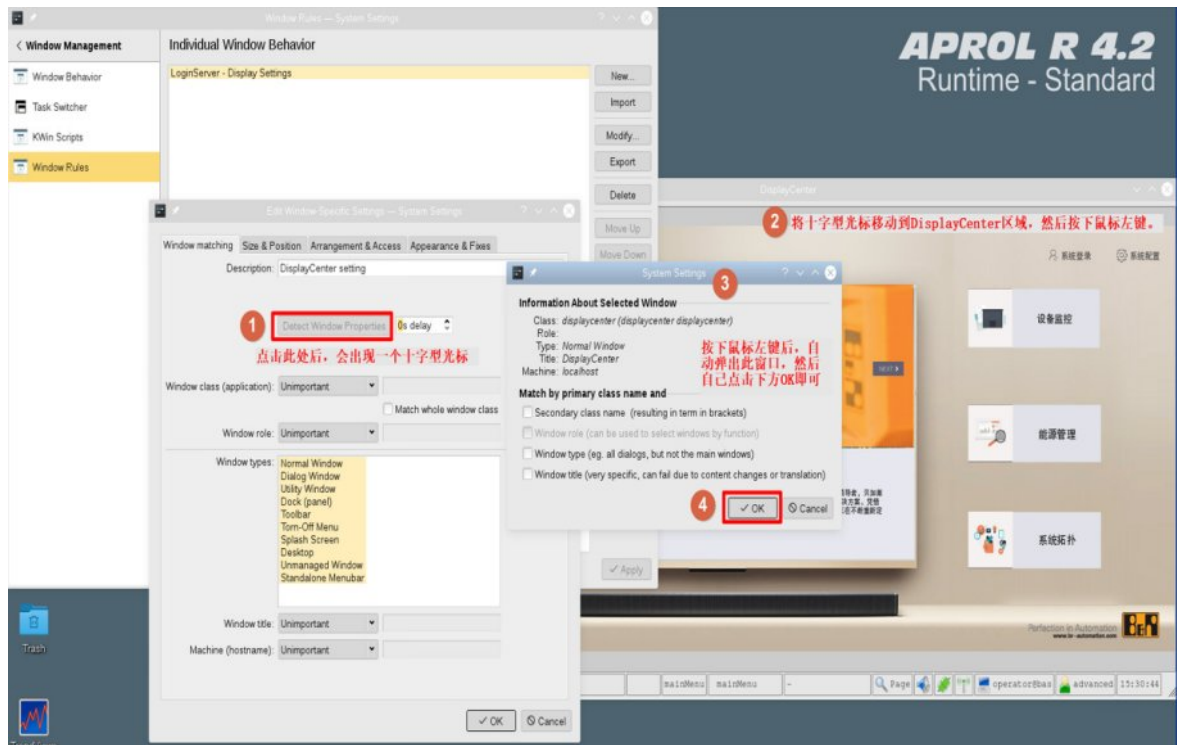
在 **DisplayCenter** 不使用 **-noWindowManager** 这个参数，那么登录相应的操作员站打开 **DisplayCenter** 后，这个 **DisplayCenter** 是带 **title bar** 无疑的。我们需要在当前的操作员站里如下面两张图示去打开 **System Settings**，开始去创建新 **DisplayCenter** 的规则。

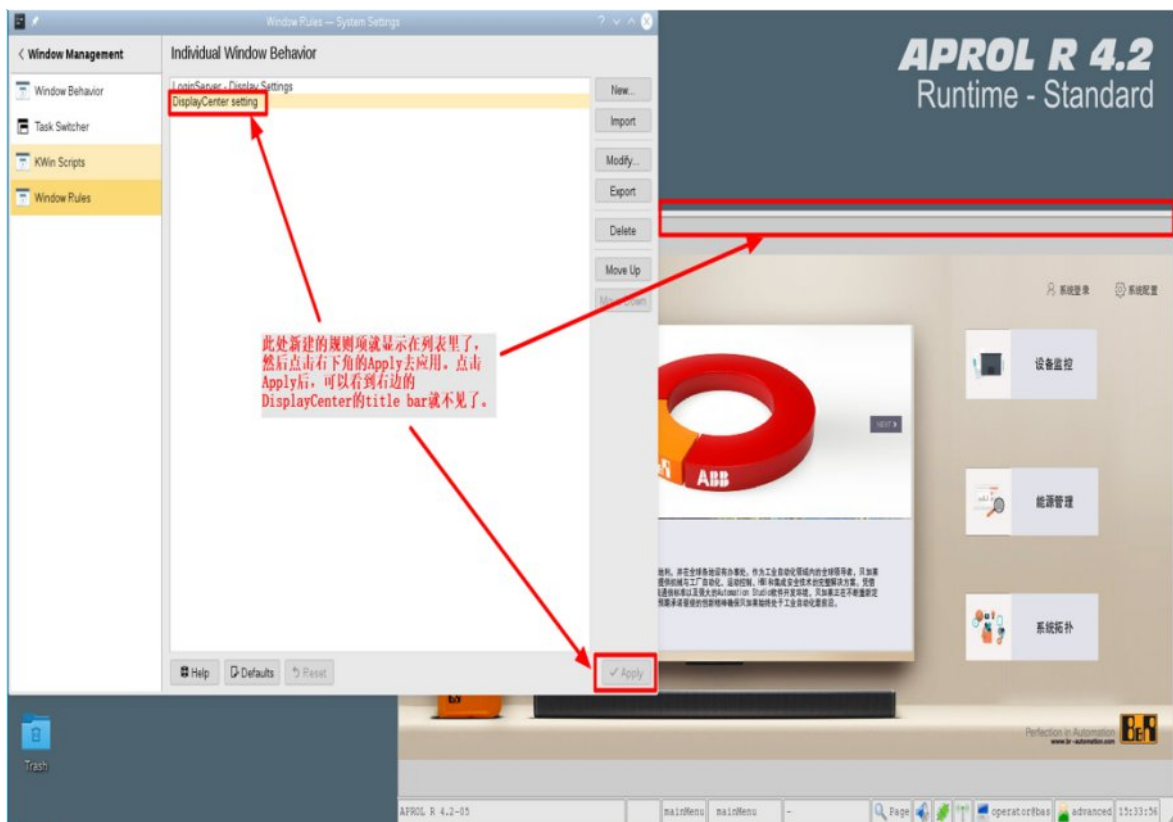
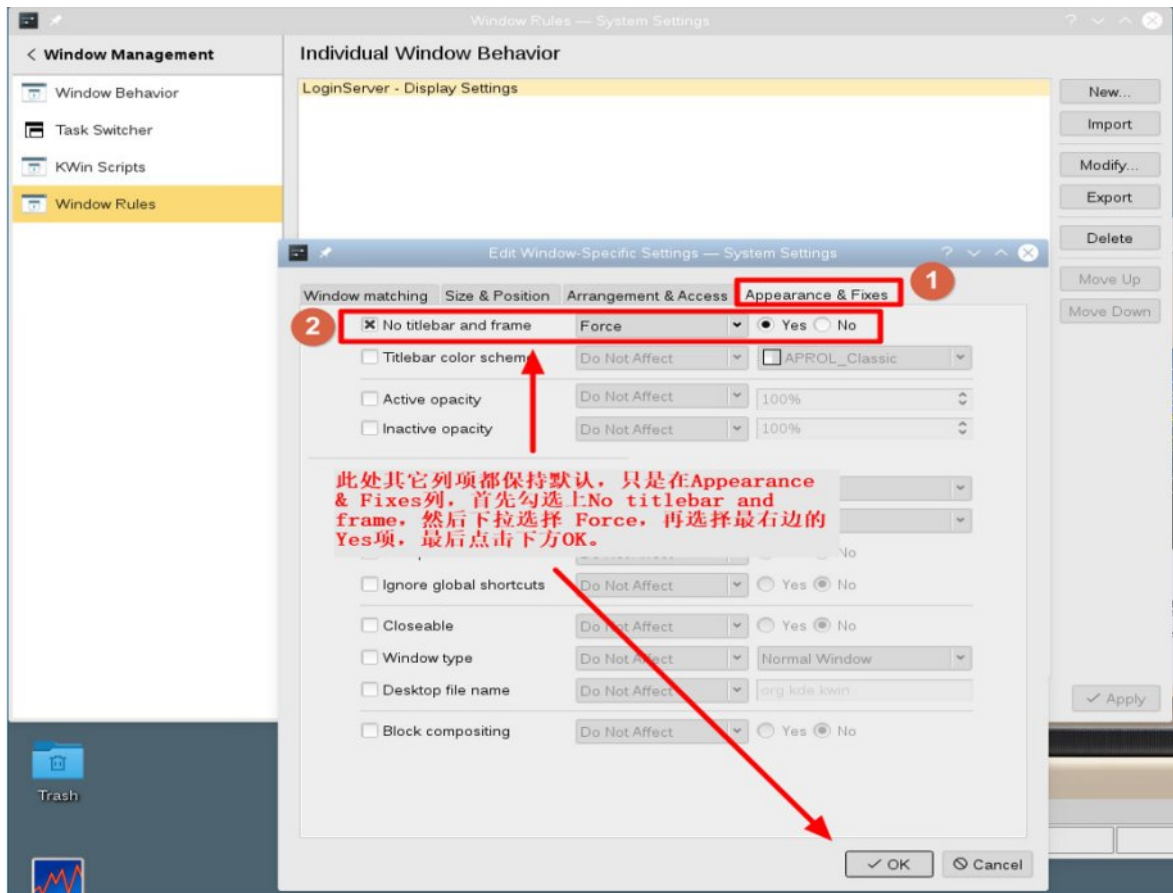




继续如下面五张图所示步骤，一步一步去执行，创建 **DisplayCenter** 的新的规则，最后应用，规则应用后的结果就是 **DisplayCenter** 不再有 **title bar** 了。但是，这样创建的规则有一个问题，就当登出当前系统重新登录或者系统重启后，创建的这个 **DisplayCenter** 的规则会消失，原因是因为系统重新登录时，后台会自动执行 **AprolUpdateKde** 脚本，这样就会把创建的 **DisplayCenter** 的规则给重置了。



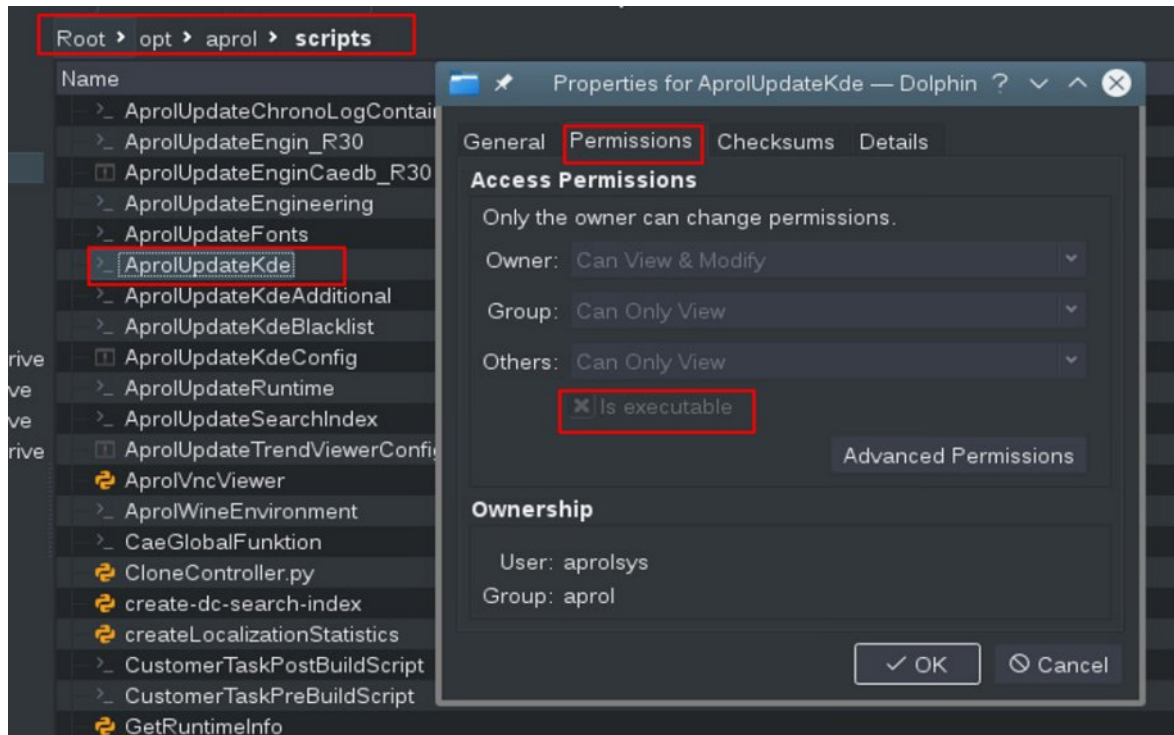




2. 禁止 ArolUpdateKde 脚本运行

我们上一节创建的 **DisplayCenter** 的新的规则由于系统登录时 **ArolUpdateKde** 脚本的执行后被重置，那我们需要去禁止 **ArolUpdateKde** 脚本文件执行，这样就可以确保我们定义的 **DisplayCenter** 的规则在系统重新登录时得以保持，从而使得 **DisplayCenter** 一直保持 **no title bar** 的状态，并且在这种情况下，测试了虚拟键盘在点击输入区域时是可以自动弹出在最前面的。

禁止 **ArolUpdateKde** 的方法很简单，即登录到本机的 **root** 系统，找到 **/opt/aprol/scripts** 目录里面的 **ArolUpdateKde** 这个脚本文件，然后如下图所示，在这个文件属性的 **Permissions** 里把 **“Is executable”** 这个勾选去掉，然后点击 **“OK”** 即可，即把这个脚本文件变成不可执行的。



⚠ 注意

ArolUpdateKde 脚本用于重置系统的桌面环境，如相关图标、菜单等等，如果因为某种原因导致桌面环境之类的破坏，可以将 **ArolUpdateKde** 脚本修改为可执行的，先执行一次，之后再重复上面从设置新 **DisplayCenter** 的规则开始的到禁止 **ArolUpdateKde** 脚本运行的整个步骤一遍即可。

· 方案选择

选择

上面提供了相应需求下的不同对应的解决方案，每种方案都有一定的局限性，请根据具体的项目需求和现场条件等综合评估后去选择最合适的一个。

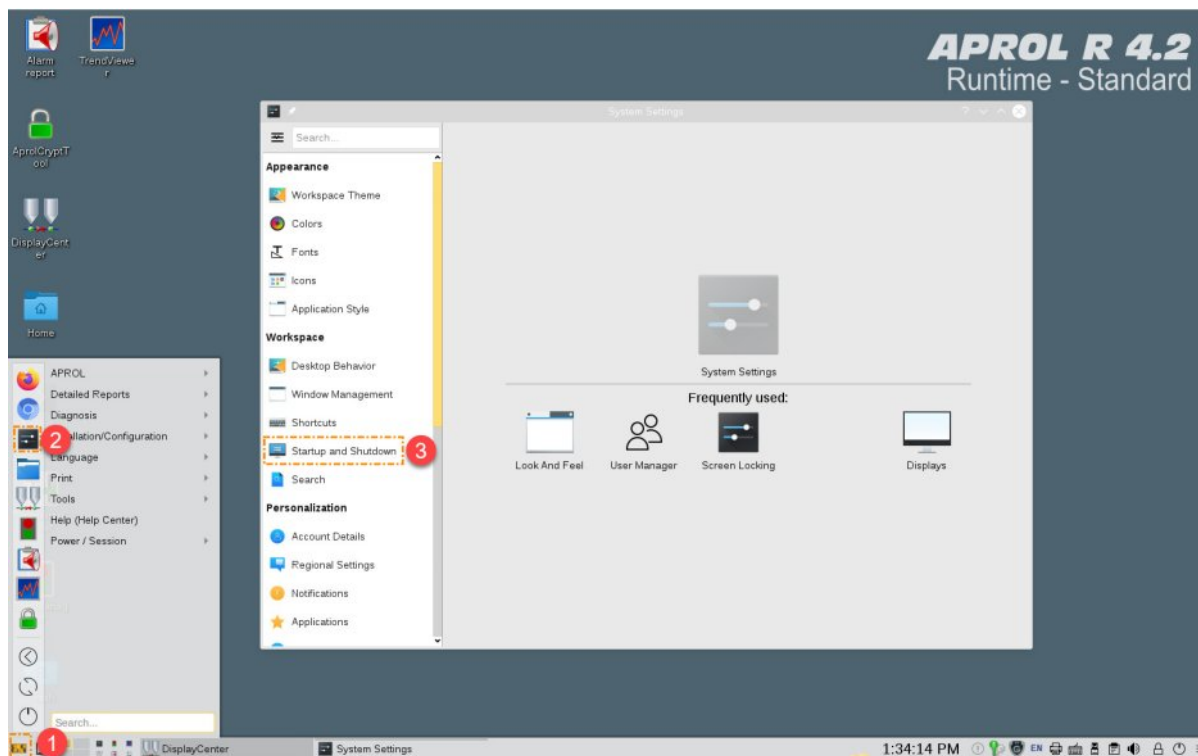
· 后续---通用方案的补充更新

采用onboard虚拟键盘方案

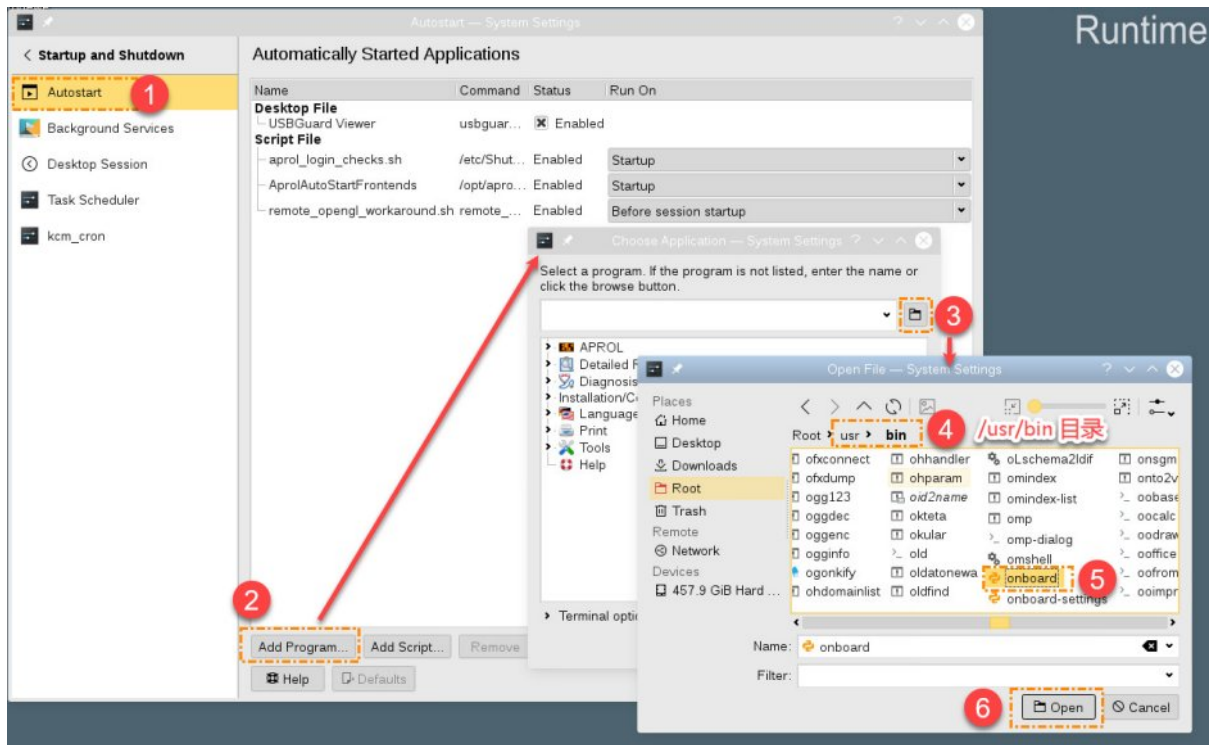
对于上面的通用方案，采用的虚拟键盘是 **xvkbd**，在某些应用下会存在一定的不便。可以尝试采用另一种虚拟键盘 **Onboard**，看是否可以满足实际项目需求。

在 APROL R4.2 里面如何去添加使用虚拟键盘 Onboard 的步骤如下，下面以登陆进 operator 系统来配置为例：

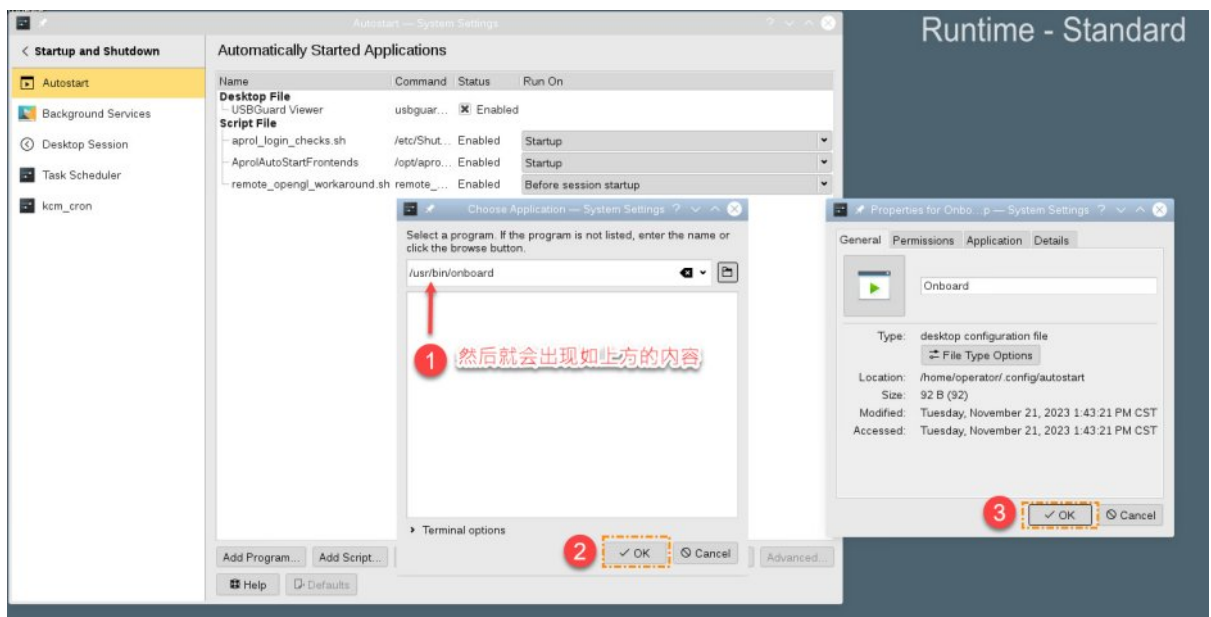
首先，如下图去打开 System Settings，点击 Startup and Shutdown，去配置虚拟键盘 Onboard 的自动启动。



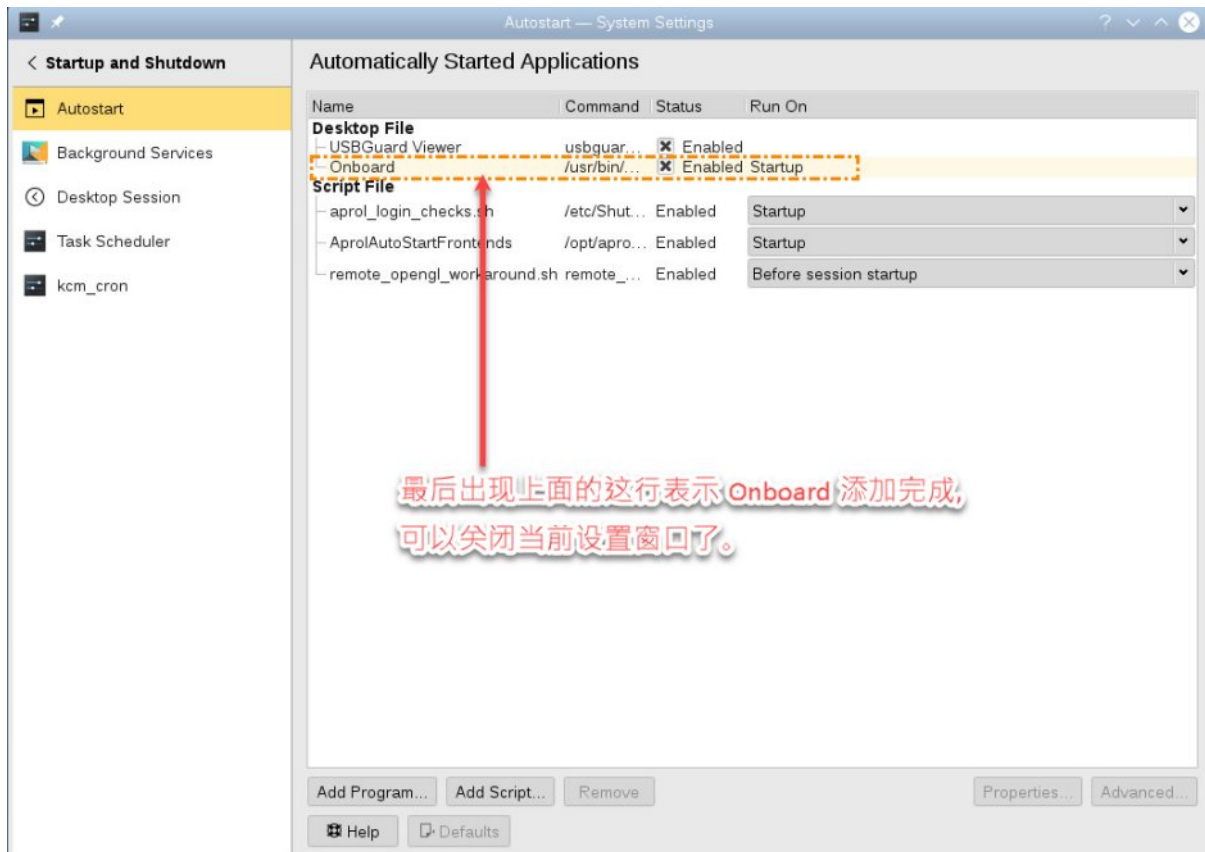
接下来按照下图的步骤去从 Startup and Shutdown 的 Autostart 里面去添加 /usr/bin 里面的 onboard 程序。



onboard 程序如下图点击 OK 去确认。



最终虚拟键盘 Onboard 成功配置成自动启动的结果如下图所示。



配置后，登出 operator 系统重新再登陆进来，就会看到如下图所示，虚拟键盘 Onboard 自动运行并显示；Onboard 不使用时可以点击去隐藏到下方状态栏，也可以点击 Onboard 的 Preference 去配置虚拟键盘的大小、样式和其它各种行为等。

