2020-6-8

庄生冷站群控模块

ZSYCLogic2 模块使用介绍说明书

杨超

上海庄生机电工程设备有限公司

目录

一、 概述

庄生冷站群控模块，为庄生公司杨超开发的 3 个模块，可适用于 Niagara 4.4 版本及以后版本，本群控模块按照 MVC 概念分为 3 个模块，这 3 个模块分别为：

1 ZSYCLogicDevice2-rt.jar ,群控模块的模型部分，即 MVC 中的 Module 部分，用来实现冷源群控中的逻辑设备的实际信息点的接入，实现设备基本逻辑，并处理来自群控逻辑的命令，将命令以正确形式发送给相应的数据点上。该模块可单独使用，配合显示模块使用更加方便。

2 ZSYCLogicView2-rt.jar 、ZSYCLogicView2-ux.jar 、ZSYCLogicView2-wb.jar ,三个 jar 文件共同对应 ZSYCLogicView2 模块，群控模块的显示部分，即 MVC 中的 View 部分，该模块可方便的提供 ZSYCLogic2 模型的视图，并对视图进行相应的配置，该模块必须配合 ZSYCLogic2 模块使用，无法单独使用。

3 ZSYCLogic2-rt.jar ，群控模块逻辑部分，即 MVC 中的 Control 部分，用来实现冷源群控中各个逻辑设备之间的互动逻辑部分，可单独使用，当然，配合本套模块中的模型模块使用更加方便。

二、 特点

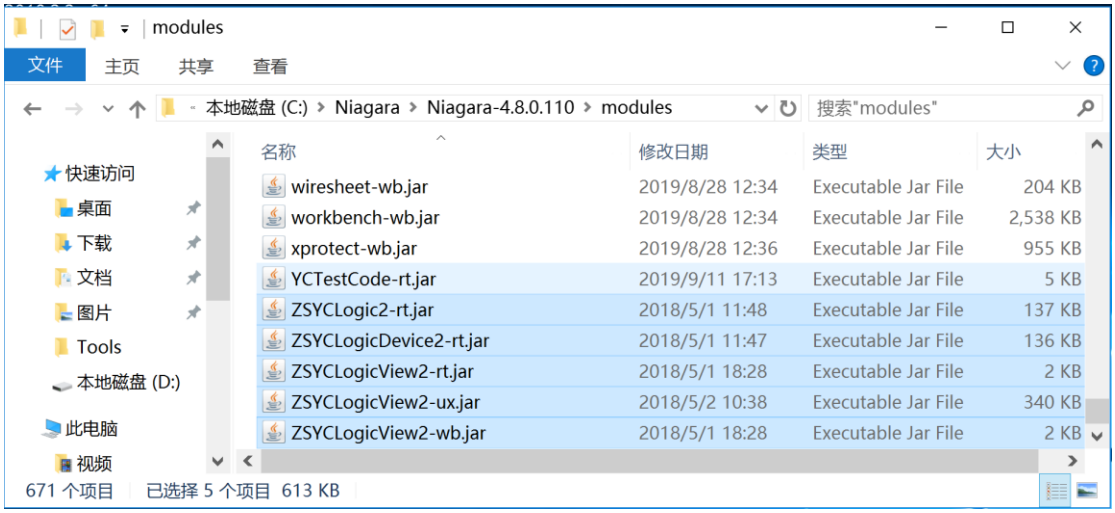
1 ZSYCLogic2 模块可方便的实现任意多并联设备的设备组轮询逻辑，可维持设备开启台数，可自动根据工作时间和优先级启动或停止设备，可方便的强制启动设备。可方便的调换设备，可列出开启队列关闭队列顺序，可列出下一个开启或关闭的设备。

2 模块可以类似流程图的形式搭建自控流程，形式上更加容易理解。

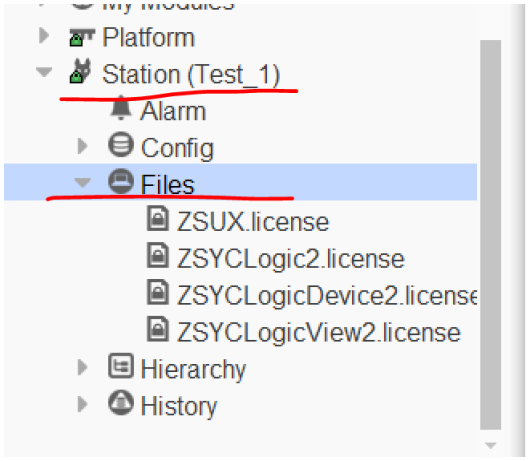
3 ZSYCLogic2 模块可方便的搭建自定义的由多种设备组成的内部含有复杂逻辑的设备群组，比如由制冷主机、阀门、板换组成的主机组等逻辑上的设备群组。

三、 准备工作

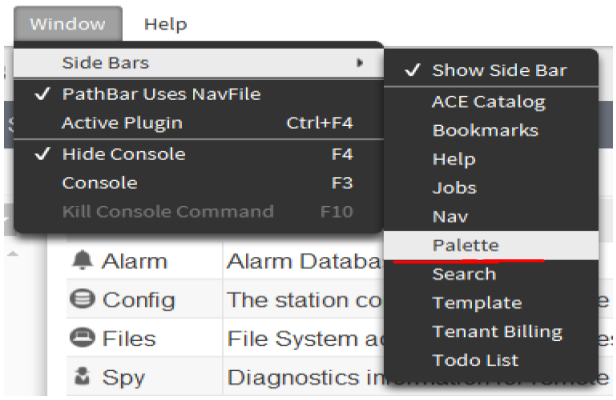
1 将模块的 jar 文件拷贝入 N4 的安装目录下的 Modules 目录下



2 将本模块的授权文件通过 N4 工作台的文件管理程序，拷贝到**相应 Station 下的 Files 目录下**。

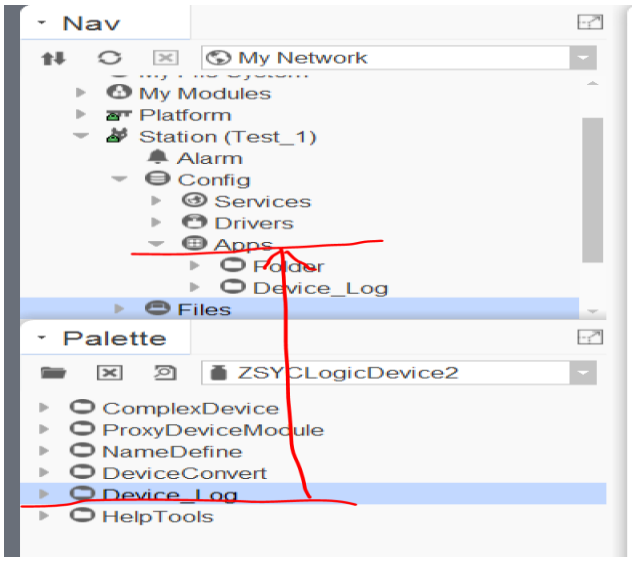


3 开启 palette 面板。



4 在 Palette 面板中打开 ZSYCLogic2 模块

5 将 Palette 面板中 ZSYCLogic2 模块下的 Logic_Log 目录整个拖放至 Station/config/Apps 目录下。

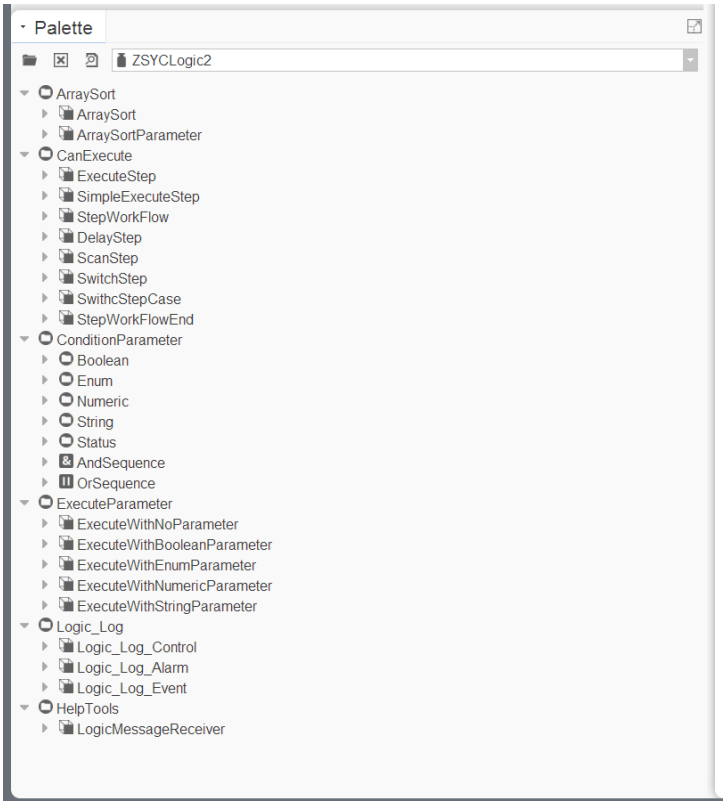


6 后期所有逻辑设备块的 3 级记录默认会写入到该文件夹下的 3 个记录辅助块中，当然，您也可以自己使用 ZSYCLogic2 模块下 HelpTools 目录下的 LogicMessageReciver 块建立自己的事件记录块。

7 重启 Station。

四、 详情

1 ZSYCLogic2 模块下所包含内容如下：



1.1 arraySort 目录：内中包括轮询逻辑块和轮询参数中介设备块。

1.2 canExecute 目录:内中包括了各种可执行步骤

1.3 ConditionParameter 目录：该目录下包括各种用于判断的条件块。

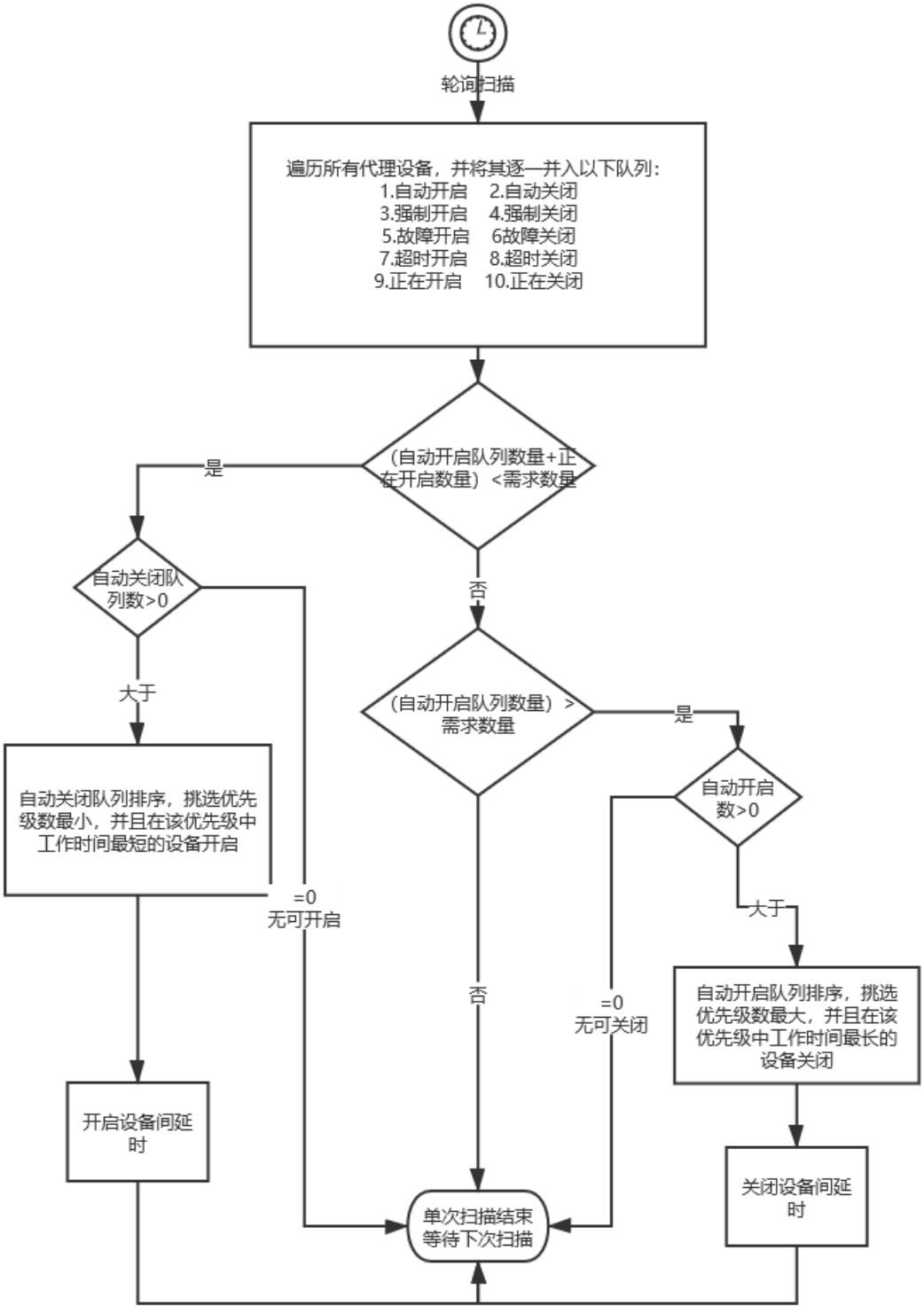
1.4 executeParameter：该目录下包括步骤执行动作参数块。

1.5 Logic_Log：该目录为预先配置好的流程信息记录目录，可整个目录拖到 Station/config/Apps 目录下使用，所有逻辑模块块的默认信息记录点路径直接指向该目录。

1.6 HelpTools：该目录下有数个帮助工具，比如统一替换路径的工具，统一执行动作的工具等等。

2 arraySort 目录详细信息:

2.1 arraySort 模块，用于对多个 arraySortParameter 中介设备块进行排序，并且根据 wantOpenUnitNum 参数值去维持相应台数的中介设备处于自动开启状态，如果系统内自动开启状态设备多了就减少，少了就增加，运行当中有故障了就新开另外设备维持系统内自动开启中介设备的台数与 wantOpenUnitNum 一致。开启或关闭中介设备时，要参考中介设备的优先级和累计工作时间。



2.1.1 logicEventName：逻辑名称，一般建议逻辑设备块的 Name 取名英文，将该逻辑的完整中文名及解释填入 logicEventName 中，该值将添加在逻辑事件日志的每条记录最开头上。

2.1.2 logicStatusMessage：逻辑当前执行信息

2.1.3 ordPath_Log_MessageReceive：逻辑事件记录对象地址。

2.1.4 logicStep：逻辑步数。正数时表示正在开启设备，负数时表示正在关闭设备。

2.1.5 isEnableAutoToStartMonitor：此值默认为 true，为 true 时，当站点重启时该逻辑模块自动开始扫描，为 false 时，station 启动后用户需要手工运行模块的 startAutoMonitor 动作才能启动扫描。

2.1.6 openDeviceDelay：连续开多台设备时，开启设备之间的延时。

2.1.7 closeDeviceDelay：连续关闭多台设备时，关闭设备之间的延时。

2.1.8 exchangeDeviceDelay：更换设备时，会先增加一台设备，再减少一台设备，这个属性定义增加一台设备后延时多长时间再减少一台设备。

2.1.9 scanDelay：扫描周期，多长时间扫描一次所有的中介设备。

2.1.10 wantOpenUnitNum：需求开启台数，此值只要更改，再下一次扫描时模块即会根据该数值调整设备组内的开启台数。

2.1.11 auto_OpenedUnitNum：设备组内自动开启的，无掉线情况，无故障情况，无超时情况，无强制控制情况的设备数量，本模块主要依据该属性和 wantOpenUnitNum 属性来调整设备开启台数。**注意**，强制开启的设备是不算在这个数量里的，比如某时刻 wantOpenUnitNum 等于 2，系统已开启 2 台设备，这时对其中一台执行“强制开启”命令，这台机器将变为强制开启状态，此时 auto_OpenedUnitNum 会变为 1，模块会自动再开启一台另外的泵。

2.1.12 auto_ClosedUnitNum：设备组内自动关闭的，无掉线情况，无故障情况，无超时情况，无强制控制情况的设备数量。

2.1.13 processing_ToOpen_UnitNum：设备组内 isProcessingToOpen 属性值为 true 的设备的数量，即设备组内正在开启的设备的数量。

2.1.14 processing_ToClose_UnitNum：设备组内 isProcessingToClose 属性值为 true 的设备的数量，即设备组内正在关闭的设备的数量。

2.1.15 erro_OpenedUnitNum:：设备组内 in_Boolean_ErrorStatus 属性值为 true，且 in_Boolean_ActiveStatus 为 true 的设备的数量，即设备组内故障且处于开启状态的设备的数量。

2.1.16 erro_ClosedUnitNum：设备组内 in_Boolean_ErrorStatus 属性值为 true，且 in_Boolean_ActiveStatus 为 false 的设备的数量，即设备组内故障且处于关闭状态的设备的数量。

2.1.17 onManual_OpenedUnitNum：设备组内 isOnManual 属性值为 true，且 in_Boolean_ActiveStatus 为 true 的设备的数量，即设备组内被强制控制且处于开启状态的设备的数量。

2.1.18 onManual_ClosedUnitNum:：设备组内 isOnManual 属性值为 true，且 in_Boolean_ActiveStatus 为 false 的设备的数量，即设备组内被强制控制且处于关闭状态的设备的数量。

2.1.19 timeOut_OpenedUnitNum：设备组内 isTimeOut 属性值为 true，且 in_Boolean_ActiveStatus 为 true 的设备的数量，即设备组内超时处于开

启状态的设备的数量。

2.1.20 timeOut_ClosedUnitNum：设备组内 isTimeOut 属性值为 true，且 in_Boolean_ActiveStatus 为 false 的设备的数量，即设备组内超时且处于关闭状态的设备的数量。

2.1.21 total_onWorkUnitNum：所有 in_Boolean_ActiveStatus 为 true 的设备的数量

2.1.22 total_closedUnitNum：所有 in_Boolean_ActiveStatus 为 false 的设备的数量

2.1.23 link_OFF_UnitNum：所有掉线设备的数量。

2.1.24 groupHaveUnitNum：群组所包含的中介设备对象的数量。

2.1.25 isOnMonitor：是否正在扫描。

2.1.26 isOnExchange：是否正在更换设备

2.1.27 nextToOpenUnit_String：下一个开启设备的说明字符串，格式为该设备的 deviceEventName_优先级_累计工作时长

2.1.28 nextToCloseUnit_String：下一个关闭设备的说明字符串，格式同上

2.1.29 nextToOpenUnit_ArrayString：开启设备的排序队列，每一个设备的说明字符串格式同上。

2.1.30 nextToCloseUnit_ArrayString：关闭设备的排序队列。

2.1.31 startAutoMonitor：开始扫描动作，执行后模块开始扫描。

2.1.32 stopAutoMonitor：停止扫描动作，执行后模块不再扫描。

2.1.33 autoOpenUnit：设定需求设备数 wantOpenUnitNum 属性的动作，参数为需求台数。

2.1.34 closeAllUnit：设定需求设备数 wantOpenUnitNum 属性为 0，自动关闭所有处于自动开启状态的设备

2.1.35 exchangeUnit：调换设备，会先多开一台设备，延时后再减少一台设备，以此达成将累计工作时间长的设备更换为累计时间短的设备。

2.2 arraySortParameter：中介设备对象，专用于 ArraySort 模块，负责接入设备的各项参数。对于中介设备对象，逻辑里预定义必须含有名字 name 为 autoOpen 和 autoClose 的 action 或可执行对象，且应该有名字 name 为 in_Boolean_ActiveStatus、in_Boolean_ErrorStatus、isTimeOut、isOK_CommunicationStatus、isOnManual、isProcessToOpen、isProcessToClose、deviceElapsedActiveTime 等七个属性或对象，如果没有请创立相应对象。

2.2.1 中介设备对象其行为应满足如下要求：

2.2.1.1 当执行 autoOpen 时，isProcessToOpen 应为 true,isProcessToClose 应为 False，执行完毕或超时后，这两个参数应被复位为都为 false。

2.2.1.2 当执行 autoClose 时，isProcessToOpen 应为 false,isProcessToClose 应为 true，执行完毕或超时后，这两个参数应被复位为都为 false。

2.2.2 ZSYCLogicDevice 里的 complexBooleanControlDevice 模块和 booleanAndFloatControlDevice 模块均完全复合上述要求可方便的被中介设备对象绑定。

2.2.3 对于由多个设备组成的复杂设备组，也可以建立文件夹，然后在文件夹中

建立 in_Boolean_ActiveStatus、in_Boolean_ErrorStatus、isTimeOut、isOK_CommunicationStatus、isOnManual、isProcessToOpen、isProcessToClose、deviceElapsedActiveTime 这几个对象，然后使用 ZSYCLogic2 里的 canExcute 文件夹下的可执行对象（建议是 stepWorkFlow 或 excuteStep）来建立 autoOpen、autoClose 动作对象，在 autoOpen 和 autoClose 里写好如何组织操作设备组里的各个设备实现开启或关闭整个设备组，再用中介代理模块绑定该文件夹，实现多个设备组合的轮询操作。

2.2.4 logicEventName：逻辑名称，一般建议逻辑设备块的 Name 取名英文，将该逻辑的完整中文名及解释填入 logicEventName 中，该值将添加在逻辑事件日志的每条记录最开头上

2.2.5 flag_DevicePriority：该中介设备的优先级，数字越小越先启动，后关闭，优先级相同的多个设备中，累计工作时间短的先开后关。

2.2.6 ord_Target_Component：中介设备对象地址，该地址下必须有名字 name 为 autoOpen 和 autoClose 的 action 或可执行对象，该地址更改时，会在该对象下查找名字 name 为 in_Boolean_ActiveStatus、in_Boolean_ErrorStatus、isTimeOut、 isOK_CommunicationStatus、isOnManual、isProcessToOpen、isProcessToClose、deviceElapsedActiveTime 等七个属性或对象，有哪个就将其相应路径填入相应的 ord_Target_属性中

2.2.7 ord_Target_in_Boolean_ActiveStatus：中介设备 in_Boolean_ActiveStatus 属性地址。

2.2.8 ord_Target_in_Boolean_ErrorStatus：中介设备 in_Boolean_ErrorStatus 属性地址。

2.2.9 ord_Target_isOK_CommunicationStatus：中介设备 isOK_CommunicationStatus 属性地址。

2.2.10 ord_Target_isOnManual：中介设备 isOnManual 属性地址。

2.2.11 ord_Target_isTimeOut：中介设备 isTimeOut 属性地址。

2.2.12 ord_Target_isProcessToOpen：中介设备 isProcessToOpen 属性地址。

2.2.13 ord_Target_isProcessToClose：中介设备 isProcessToClose 属性地址。

2.2.14 ord_Target_deviceElapsedActiveTime：中介设备 deviceElapsedActiveTime 属性地址。

2.2.15 in_Boolean_ActiveStatus：中介设备 in_Boolean_ActiveStatus 属性，标志中介设备的运行状态。

2.2.16 in_Boolean_ErrorStatus：中介设备 in_Boolean_ErrorStatus 属性地址。标志中介设备的故障状态。

2.2.17 isOK_CommunicationStatus：中介设备 isOK_CommunicationStatus 属性，标志中介设备的通讯状态

2.2.18 isOnManual：中介设备 isOnManual 属性，标志中介设备是否被强制控制

2.2.19 isTimeOut 中介设备 isTimeOut 属性，标志中介设备是否操作超时。

2.2.20 isProcessToOpen：中介设备 isProcessToOpen 属性，标志中介设备是否正在开启。

2.2.21 isProcessToClose：中介设备 isProcessToClose 属性，标志中介设备是否正在关闭。

2.2.22 deviceElapsedActiveTime：中介设备 deviceElapsedActiveTime 属性，记录中介设备的运行时间累计时长。

2.2.23 autoLink：自动绑定动作，从 Ord_Target 指定的各个目标地址链接到相应的属性中。

2.2.24 autoOpen：自动开启命令，将执行 ord_Target_Component 指向的对象下的 name 名为 autoOpen 的 action 或可执行对象。

2.2.25 autoClose：自动关启命令，将执行 ord_Target_Component 指向的对象下的 name 名为 autoClose 的 action 或可执行对象。

3 CanExecute，可执行块目录，该目录下的块可近似看作流程图中执行、选择、循环的对应，其详细如下：

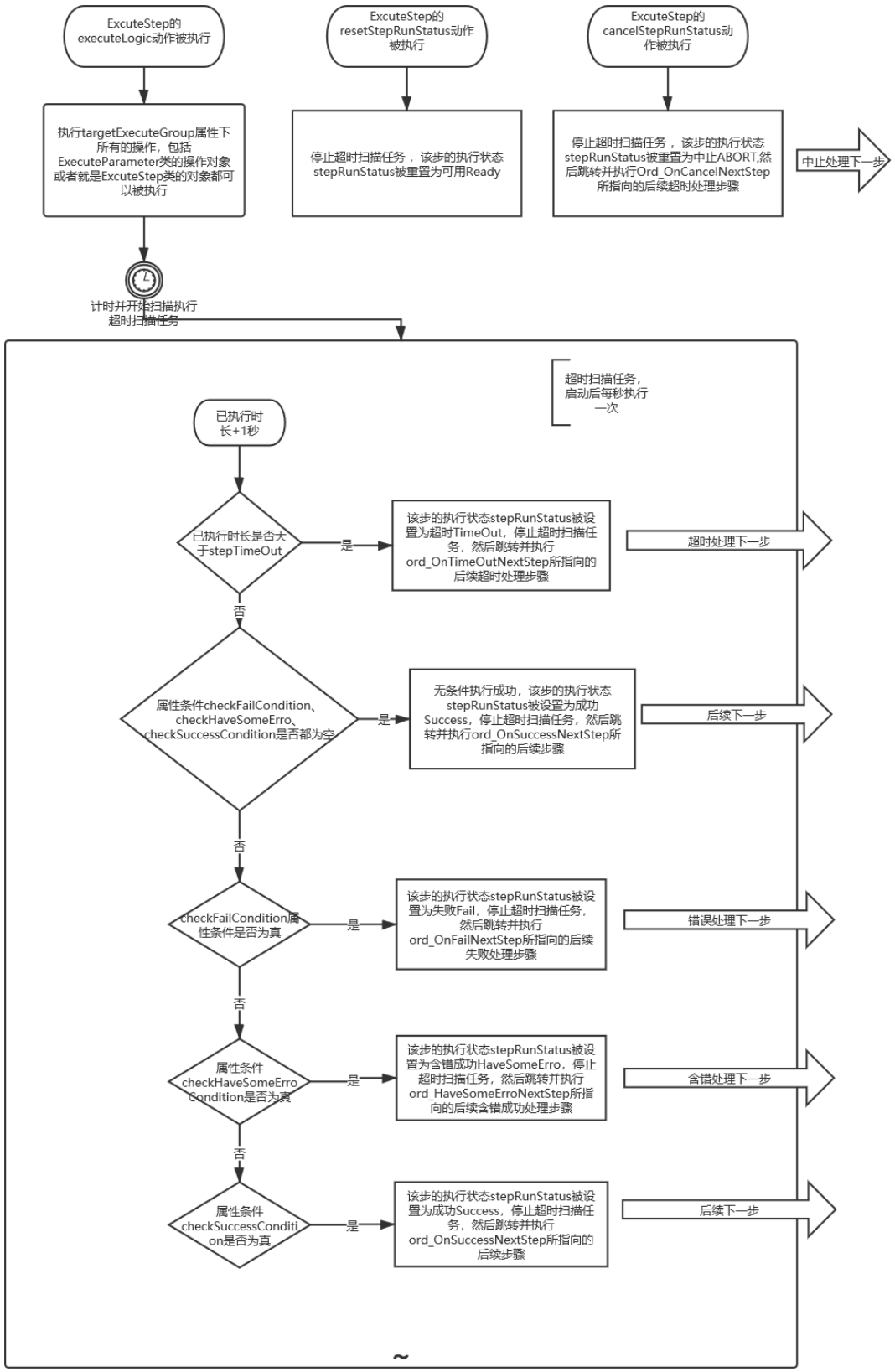
3.1 该目录下所有块均含有 executeLogic 动作，可以执行该模块的主动作，相互之间可以直接调用。

3.2 该目录下所有块均含有 resetStepRunStatus 动作，可以执行该模块的复位动作。

3.3 该目录下所有块均含有 cancelStepRunStatus 动作，可以执行该模块的中断执行动作

3.4 该目录下各个块指向的下一步，都可指向 CanExcute 对象，如果是 canExcute 对象则在跳转时会直接执行该对象的 executeLogic 对象。一般来说建议指向 ExcuteStep 或者 StepWorkFlow 对象，如果是最后一步无跳转无条件检测只执行不关心结果，则建议是 SimpleExecuteStep。

3.5 ExecuteStep：带结果判断的执行步骤



3.5.1 stepRunStatus：标识逻辑模块当前状态，有如下几种状态

3.5.1.1 0 - Ready ，可用状态

3.5.1.2 1 - Success ，逻辑执行成功

3.5.1.3 2 - Processing ，逻辑正在执行

3.5.1.4 3 - TimeOut ，逻辑执行充实

3.5.1.5 4 - Abort ，逻辑执行中止

3.5.1.6 5 - Fail ，逻辑执行失败

3.5.1.7 6 - HaveSomeErro ，逻辑执行成功，但执行过程中有错误

3.5.2 stepTimeOut：该步逻辑的超时时间，若该模块执行逻辑后，几个条件判断不全都为空，并且未达到任何一个条件为真，并且累计执行时间又超过这个时间，即被判定为超时，模块将中止执行并跳转至超时处理后续逻辑，模块的状态将被设定为 TimeOut。

3.5.3 targetExecuteGroup：该步骤的主要执行块，当 ExcuteStep 的 executeLogic 动作被执行时，将首先执行 targetExecuteGroup 属性下所有的操作，包括 ExecuteParameter 类的操作对象或者就是 canExcute 类的对象都可以被执行。用户需自己手动将 ExecuteParameter 对象或 canExcute 类的对象拖放到该对象下并进行配置。

3.5.4 checkFailCondition：失败条件检查，该对象是一个与条件队列，其下的所有条件为真时才为真，用户需要手动拖入 ConditionParameter 下的判断条件参数来进行判断。若该条件检查值为 true，该步的执行状态 stepRunStatus 被设置为失败 Fail，停止超时扫描任务，然后跳转并执行 ord_OnFailNextStep 所指向的后续失败处理步骤。

3.5.5 checkHaveSomeErroCondition：含错成功条件检查，该对象是一个与条件队列，其下的所有条件为真时才为真，用户需要手动拖入 ConditionParameter 下的判断条件参数来进行判断。若该条件检查值为 true，该步的执行状态 stepRunStatus 被设置为含错成功 HaveSomeErro，停止超时扫描任务，然后跳转并执行 ord_HaveSomeErroNextStep 所指向的后续含错处理步骤。

3.5.6 checkSuccessCondition：成功条件检查，该对象是一个与条件队列，其下的所有条件为真时才为真，用户需要手动拖入 ConditionParameter 下的判断条件参数来进行判断。若该条件检查值为 true，该步的执行状态 stepRunStatus 被设置为成功 Success，停止超时扫描任务，然后跳转并执行 ord_OnSuccessNextStep 所指向的后续步骤

3.5.7 ord_OnFailNextStep：指向后续失败处理步骤对象。该对象必须是 canExcute 类的对象。

3.5.8 ord_HaveSomeErroNextStep：指向后续含错成功处理步骤对象。该对象必须是 canExcute 类的对象。

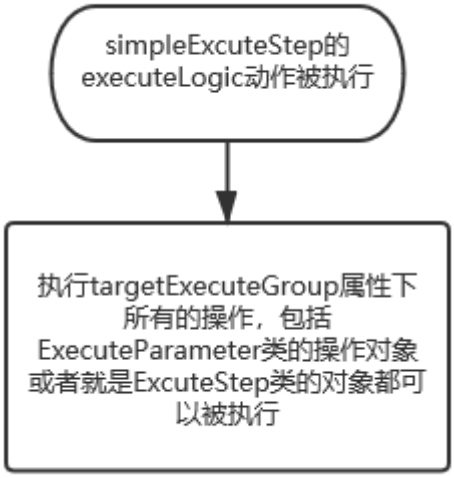
3.5.9 ord_OnSuccessNextStep：指向成功后续处理步骤对象。该对象必须是 canExcute 类的对象。

3.5.10 ord_OnTimeOutNextStep：指向超时后续处理步骤对象。该对象必须是 canExcute 类的对象。

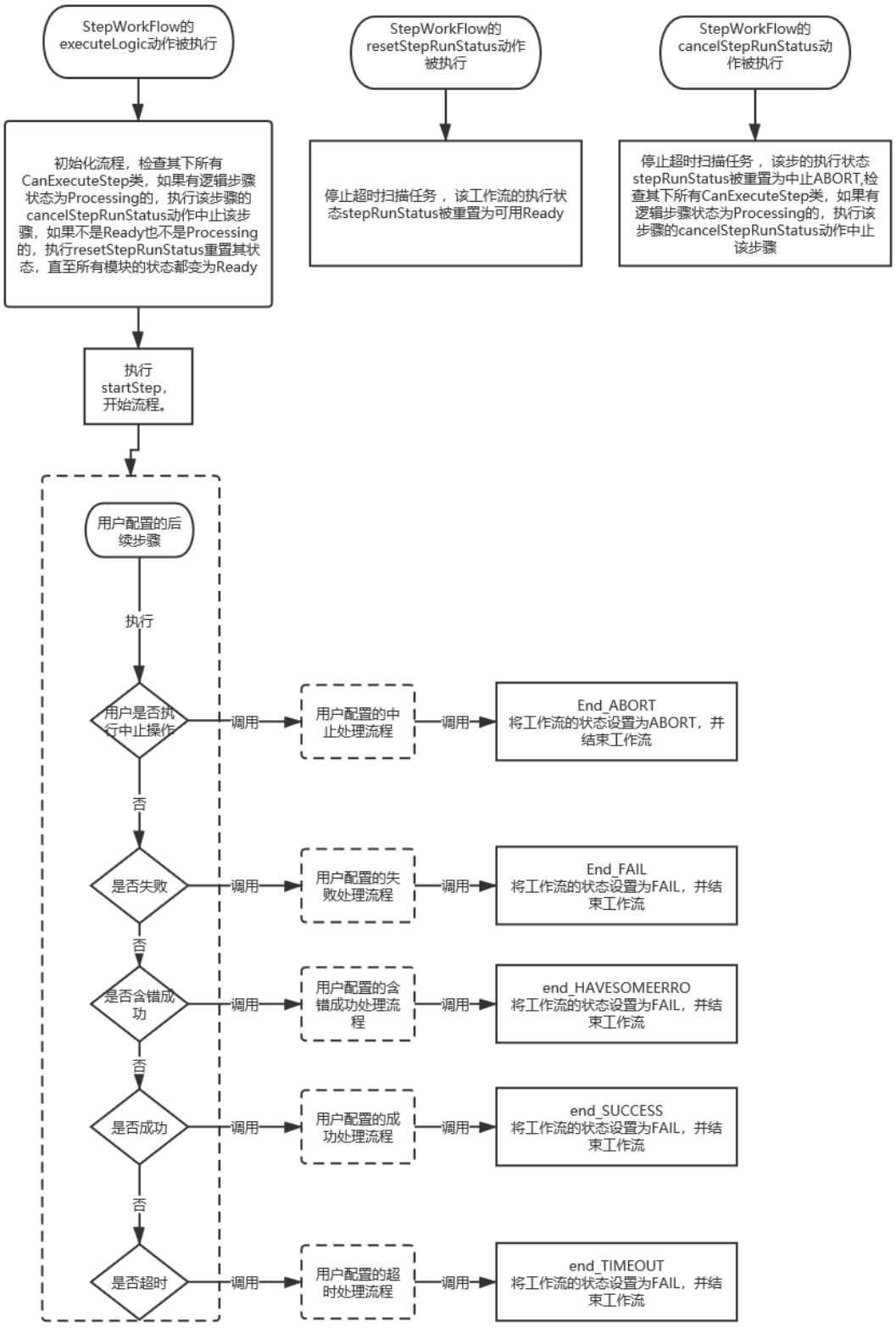
3.5.11 ord_OnCancelNextStep：指向中止后续处理步骤对象。该动作将在 cancelStepRunStatus 动作执行后被执行，该对象必须是 canExcute 类的对象。

3.6 SimpleExecuteStep：简单步骤，只执行操作，不检查结果，不跳转下一步。

3.6.1 targetExecuteGroup：该步骤的主要执行块，当 ExcuteStep 的 executeLogic 动作被执行时，将首先执行 targetExecuteGroup 属性下所有的操作，包括 ExecuteParameter 类的操作对象或者就是 canExcute 类的对象都可以被执行。用户需自己手动将 ExecuteParameter 对象或 canExcute 类的对象拖放到该对象下并进行配置。



3.7 StepWorkFlow：工作流，代表一个可执行一次的，含有多个步骤的流程，该流程可标识是否执行成功或失败，该流程默认含有一个名为 startStep 的静态对象 excuteStep 类对象作为开始步骤，后续的步骤都由该步骤跳转。



3.7.1 stepRunStatus：工作流的状态，同本文 3.5.1

3.7.2 workFlowTotalTimeOut：工作流整体执行超时时间，该时间理论上最好要大于工作流其下各个 excuteStep 的 timeout 之和。

3.7.3 startStep：默认静态属性，工作流的开始步骤，后续步骤需用户从 canExcute 目录下拖 excuteStep 出来到 workFlowStep 下，并使用 startStep 来指向它。

3.7.4 end_SUCCESS：成功结尾，用户配置的流程在执行成功后，应该指向并执行该结尾。该结尾将会结束工作流并将工作流状态置为 SUCCESS。

3.7.5 end_FAIL：失败结尾，用户配置的流程在执行失败后，应该指向并执行该结尾。该结尾将会结束工作流并将工作流状态置为 FAIL。

3.7.6 end_ABORT：中止结尾，用户配置的流程步骤在执行中止后，应该指向并执行该结尾。该结尾将会结束工作流并将工作流状态置为 ABORT。

3.7.7 end_HAVESOMEERRO：含错成功结尾，用户配置的流程在执行含错成功后，应该指向并执行该结尾。该结尾将会结束工作流并将工作流状态置为 HAVESOMEERRO。

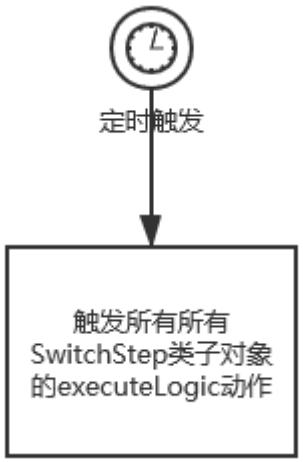
3.7.8 end_TIMEOUT：超时结尾，用户配置的流程在执行超时后，应该指向并执行该结尾。该结尾将会结束工作流并将工作流状态置为 TIMEOUT。

3.8 DelayStep：该步骤的 executeLogic 动作执行时，会延时属性 delayTime 规定的延时时长，并在延时到了以后跳转至 ord_OnSuccessNextStep 指向的下一步步骤去。

3.8.1 delayTime：延时时长

3.8.2 ord_OnSuccessNextStep：下一步目标步骤地址

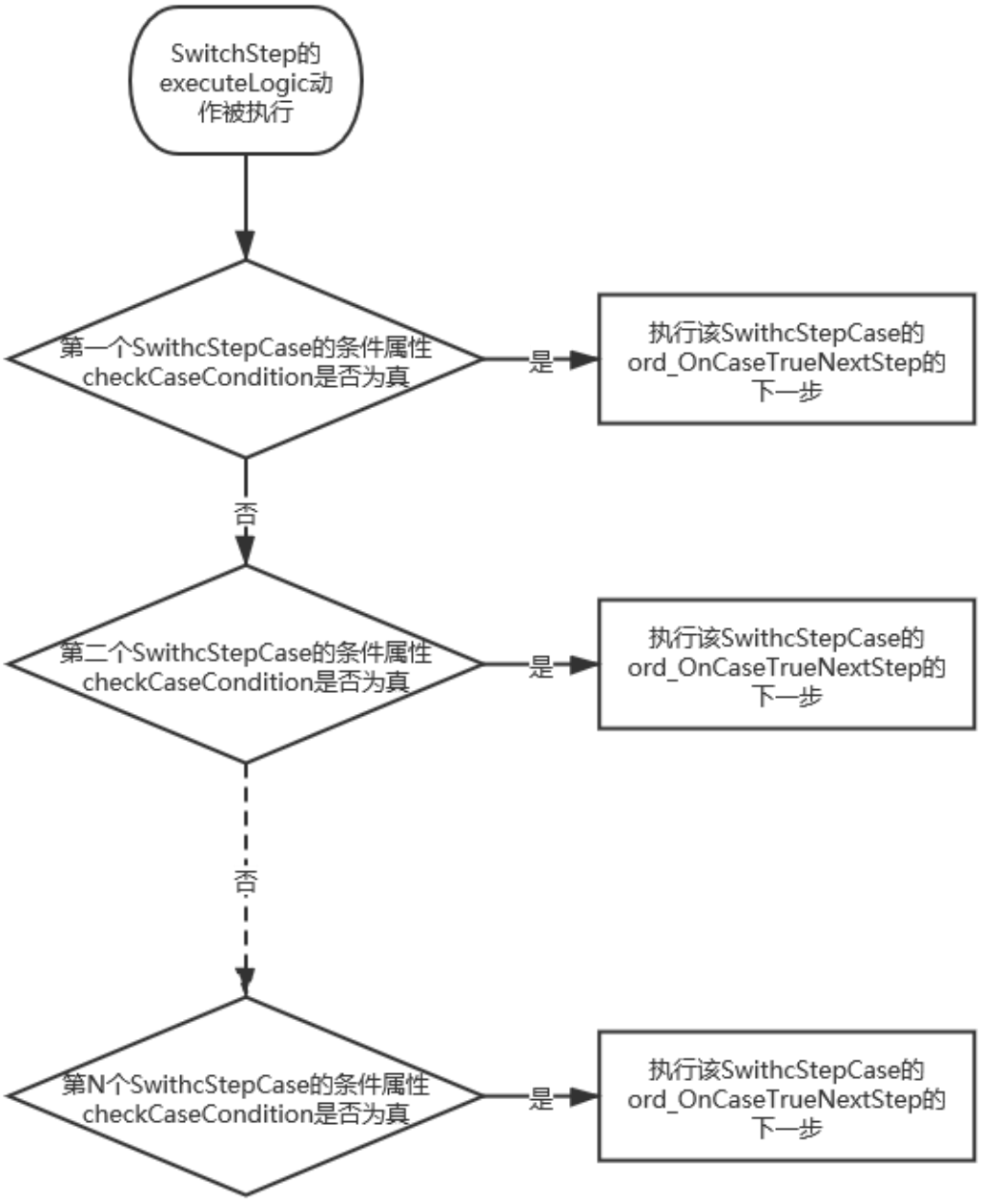
3.9 ScanStep：定时扫描步骤，将按照 scanDelay 指定的间隔时间，执行 ScanStep 下的所有 SwitchStep 类对象的 executeLogic 动作（默认有一个静态的名为 targetSwitchStep 的 SwitchStep 对象，用户也可按照需要添加其他的 SwitchStep 类对象，各个 SwitchStep 类对象之间是平级关系），SwitchStep 类对象的 executeLogic 动作按照顺序检查该 SwitchStep 类对象下的各个 SwithcStepCase 里的条件，哪个 SwithcStepCase 里的条件为 true，就执行该 SwithcStepCase 指向的后续步骤，并不再检查该 SwitchStep 类对象下的其他 SwithcStepCase。



3.9.1 scanDelay：扫描间隔

3.9.2 targetSwitchStep：默认静态执行选择器

3.10 SwitchStep：执行选择器，SwitchStep 类对象的 executeLogic 动作按照顺序检查该 SwitchStep 类对象下的各个 SwithcStepCase 里的条件，哪个 SwithcStepCase 里的条件为 true，就执行该 SwithcStepCase 指向的后续步骤，并不再检查该 SwitchStep 类对象下的其他 SwithcStepCase。



3.11 SwithcStepCase：SwitchStep 下的条件配置项

3.11.1 checkCaseCondition：检查条件

3.11.2 ord_OnCaseTrueNextStep：条件为真时需要执行的步骤的目标地址。

3.12 StepWorkFlowEnd：StepWorkFlow 工作流对象的结束对象，执行该对象的 executeLogic 动作，将会把其所属的 StepWorkFlow 工作流对象的状态设置为其属性 workFlowResult_RunStatus 所指定的状态。

4 ConditionParameter：判断条件参数对象

4.1 主要用于 canExcute 对象和 switchCase 里的各项判断。

4.2 判断条件参数对象是个布尔量，它都有一个隐藏属性 checkValue，标识该条件判断后的结果是 true 还是 false。

4.3 根据数据类型分为 Boolean、Enum、Numeric、String、Status 几种判断条件

4.4 有与队列和或队列两种队列，对应有多个条件需要与和或操作时的需求，与队列和或队列可嵌套。

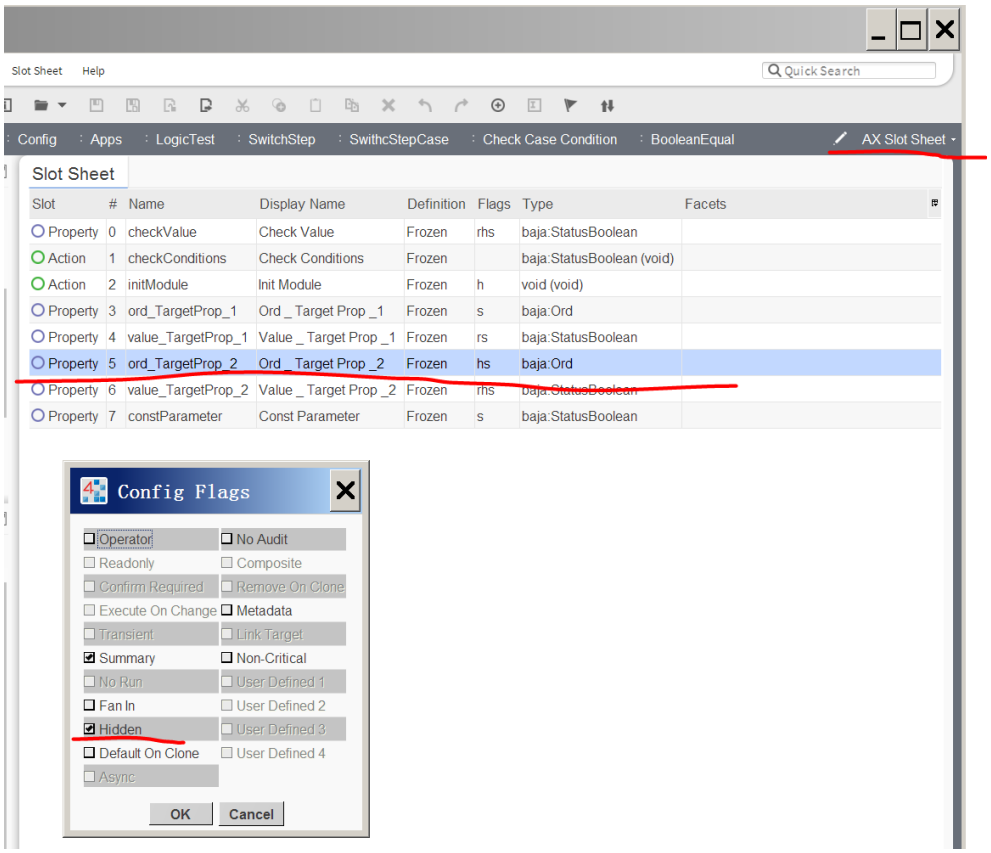
4.5 参数对象都有 checkConditions 动作，当执行该动作时将检查判断条件参数对象的各个参数值，并得出该参数对象的值为 false 还是 true。

4.6 Boolean、Enum、Numeric、String 都有如下 5 个属性：

4.6.1 ord_TargetProp_1：用来比较的属性的目标地址

4.6.2 value_TargetProp_1：用来比较的属性的目标对象的输出值，只读，用户不可更改此值，当 ord_TargetProp_1 正确时，该值将在 checkConditions 动作执行时自动更新。**注意，该值并不时刻保持更新，只在 checkConditions 动作执行时或 ord_TargetProp_1 变动时更新。**

4.6.3 ord_TargetProp_2：用来与 ord_TargetProp_1 比较的属性的目标地址，默认隐藏，如需使用，请在 ConditionParameter 对象上右键单击，选择 views/axSlotSheet 视图，并在该视图中该属性上右键单击，选择 config Flags，将其的 Hidden 标志勾选掉才能用。如果该值不为空 null，则该判断条件会使用该地址指向的目标点输出值是否和 ord_TargetProp_1 的目标点输出值进行比较。



4.6.4 value_TargetProp_2：ord_TargetProp_2 对应的对象的输出值，默认隐藏，特性同上。

4.6.5 constParameter：比较值，如果 ord_TargetProp_2 值为空 null，则该判断条件会使用该地址指向的目标点输出值是否和 constParameter 值进行比较

4.7 Boolean：布尔量的比较条件，其 constParameter 为布尔量

4.7.1 BooleanEqual：布尔等于，判断参与比较的量是否等于，相等时为 true

4.7.2 BooleanNotEqual：布尔不等于，判断参与比较的量是否不相等，不相等时为真

4.7.3 BooleanAnd：布尔与，参与比较的量都为 true 时为 true，其他为 false

4.7.4 BooleanOr：布尔或，参与比较的量只要一个为 true 时就为 true，都为 false 时才为 false。

4.8 Enum：枚举量检测，注意该检测是检测目标枚举量的输出值的 Ordinal 数值，两个 enum 的 Ordinal 数值相同即为相同，并**不去检测**枚举量的 Tag 量。

4.8.1 EnumEqual：检测是否相等,相等时为 true

4.8.2 EnumNotEqual：检测是否不等，不相等时为 true.

4.9 Numeric：数字量检测

4.9.1 NumericLessOrEqual：小于等于

4.9.2 NumericNotEqual：不等

4.9.3 NumericGreater：大于

4.9.4 NumericGreaterOrEqual：大于等于

4.9.5 NumericLess：小于

4.9.6 NumericEqual：等于

4.10 String：字符量检测

4.10.1 StringNotEqual：是否不等

4.10.2 StringEqual：等于

4.11 Status：检测状态量的状态。

4.11.1.1 StatusCheck：检测状态量的状态，该对象只有一个 ord_TargetProp_1，没有隐藏的 ord_TargetProp_2

4.11.1.2 ord_TargetProp_1：需要检测的状态量，该量必须是 BStatus 的子类，比如 BBooleanWritable、BBooleanPoint 等等，也可以是 BBooleanWritable 下的 in10 等等 BStatusBoolean 类的属性。

4.11.1.3 check_Status_Selector：要检测的状态：

- "IS_OK"：检测点是否是正常状态，正常时为 true
- "IS_NOT_OK"：检测点是否处于非正常状态，不正常时为 true
- "IS_ALARM"：检测点是否报警，报警时为 true
- "IS_DISABLED"：检测点是否被禁用,被禁用时为 true
- "IS_DOWN"：检测点是否掉线，掉线时为 true
- "IS_NULL"：检测点输出值是否是空 NULL，是 NULL 时为 true
- "IS_FAULT":检测点通讯是否错误，错误时为 true
- "IS_OVERRIDDEN":检测点是否被强制

4.12 AndSequence：与队列，其下所有 ConditionParameter 类的子对象的值都为 true 时，该队列的值才为 true，其他情况均为 false。

4.13 OrSequence：或队列，其下所有 ConditionParameter 类的子对象的值都为 false 时，该队列的值才为 false，任意一个 ConditionParameter 类的子对象的值为 true 时该队列的值为 true。

5 ExecuteParameter：执行参数，将执行指定对象的动作，根据指定对象的类型不同，执行的动作有差别

5.1 都有一个 parameter_value 属性，指定执行该动作的参数。

5.2 BExecuteWithNoParameter 的 parameter_value 属性无用，被隐藏。

5.3 可指定的对象包括：

5.3.1 BCanExecuteStep：执行 BCanExecuteStep 对象的 executeLogic 动作，该动作无需参数。

5.3.2 BBooleanWritable：需 ExecuteWithBooleanParameter 对象来执行，该动作的执行将会调用 BooleanWritable 点 set 动作，参数为 parameter_value 属性指定的 boolean 值，该动作将会更改 BooleanWritable 的 fallback 输入值，如果该点未被强制或其他高优先级输入抢占，则其 out 值会被更改。

5.3.3 BEnumWritable：需 ExecuteWithEnumParameter 对象来执行，该动作的执行将会调用 EnumWritable 点 set 动作，参数为 parameter_value 属性指定的 Ordinal 数值，该动作将会更改 EnumWritable 的 fallback 输入值，如果该点未被强制或其他高优先级输入抢占，则其 out 值会被更改。

5.3.4 BNumericWritable 需 ExecuteWithNumericParameter 对象来执行，该动作的执行将会调用 NumericWritable 点 set 动作，参数为 parameter_value 属性指定的值，该动作将会更改 NumericWritable 的 fallback 输入值，如果该点未被强制或其他高优先级输入抢占，则其 out 值会被更改。

5.3.5 BStringWritable 需 ExecuteWithStringParameter 对象来执行，该动作的执行将会调用 StringWritable 点 set 动作，参数为 parameter_value 属性指定的值，该动作将会更改 StringWritable 的 fallback 输入值，如果该点未被强制或其他高优先级输入抢占，则其 out 值会被更改。

5.3.6 BStatusBoolean：需 ExecuteWithBooleanParameter 对象来执行，该动作的执行将会将目标点设为 parameter_value 属性指定的值。

5.3.7 BstatusEnum：需 ExecuteWithEnumParameter 对象来执行，该动作的执行将会将目标点设为 parameter_value 属性指定的值。

5.3.8 BDynamicEnum：需 ExecuteWithEnumParameter 对象来执行，该动作的执行将会将目标点设为 parameter_value 属性指定的值。

5.3.9 BFrozenEnum：需 ExecuteWithEnumParameter 对象来执行，该动作的执行将会将目标点设为 parameter_value 属性指定的值。

5.3.10 BStatusNumeric：需 ExecuteWithNumericParameter 对象来执行，该动作的执行将会将目标点设为 parameter_value 属性指定的值。

5.3.11 BStatusString：需 ExecuteWithStringParameter 对象来执行，该动作的执行将会将目标点设为 parameter_value 属性指定的值。

5.3.12 Action：执行目标地址所指向的动作。参数为 parameter_value 属性指定的值。

5.4 ExecuteWithNoParameter：无参数动作

5.5 ExecuteWithBooleanParameter：带布尔参数的动作

5.6 ExecuteWithEnumParameter：带枚举参数的动作

5.7 ExecuteWithNumericParameter：带浮点量参数的动作

5.8 ExecuteWithStringParameter：带字符串参量的动作

6 Logic_Log 目录：该目录为预先配置好的逻辑信息记录目录，可整个目录拖到 Station/config/Apps 目录下使用，所有逻辑块的默认信息记录点路径直接指向该目录。

7 HelpTools 目录：

7.1 LogicMessageReceiver：自定义逻辑信息记录点。