

## 点 (Point) 的 facets

基本上, facets 决定了 point 的值如何在 station 上显示。下面的例子包括 numeric 类型的工程单位 (engineering unit) 和小数精度 (decimal precision), 以及 Boolean 和 enum 类型的描述性值 (状态) 的文本。

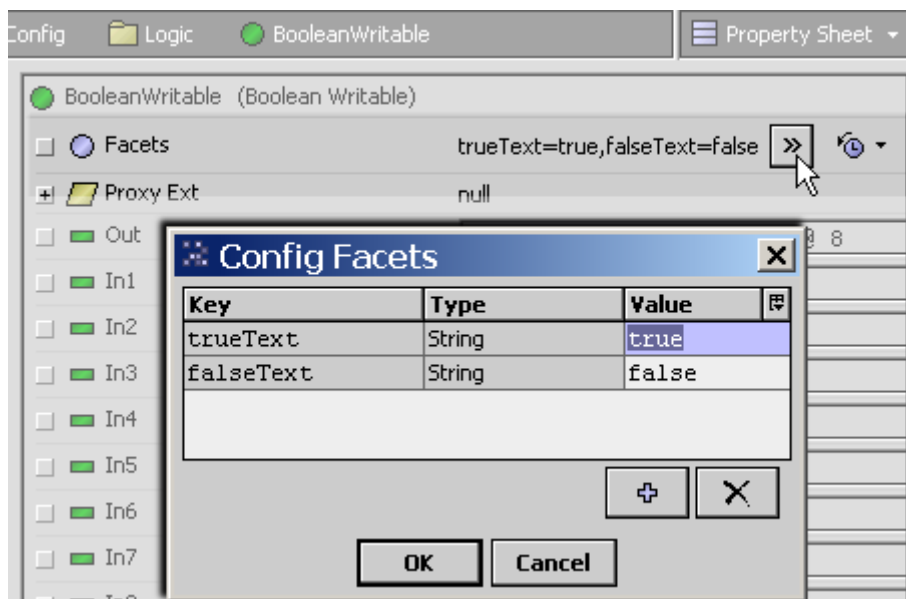
除了 proxy points (可能定义了设备的 facets) 以外, point facets 不影响 point 值的处理。更多详细信息, 请参考“Niagara 驱动指南”。

**注意:** 除了 control points 以外, 各种其它组件也有 facets。例如, 许多 kitControl 和 schedule 组件就有 facets。除非特别说明, 有关 point facets 的细节也适用于这些组件。

### 访问和编辑 facets

Facets 是 kitControl 模块中所有 control points 和对象的一个 frozen slot。如下图所示, 您可以在 point 的 property sheet 中使用 **Config Facets** 对话框修改 facets 值。

图 79 Point facets 及其编辑对话框



在这种情况下, BooleanWritable facet 的默认 trueText 为“true” – 如要修改, 您只需简单地点击进行选择, 然后输入任何需要的文本即可。例如, 将“true”修改为“On”, 将“false”修改为“Off”。完成操作后点击 **OK**, 然后点击 **Save**, 就完成了对 point 的实际修改。

有时, 除了修改现有的 facets 以外, 还可以在 point 上添加或删除 facets 值。在许多 points 上, 可能只需修改默认的 facets 或为默认 facets 提供新值。如果是 writable points, 还可以添加一个 facet 去限制覆盖的时间长度 (参考“Maximum override duration face”)。

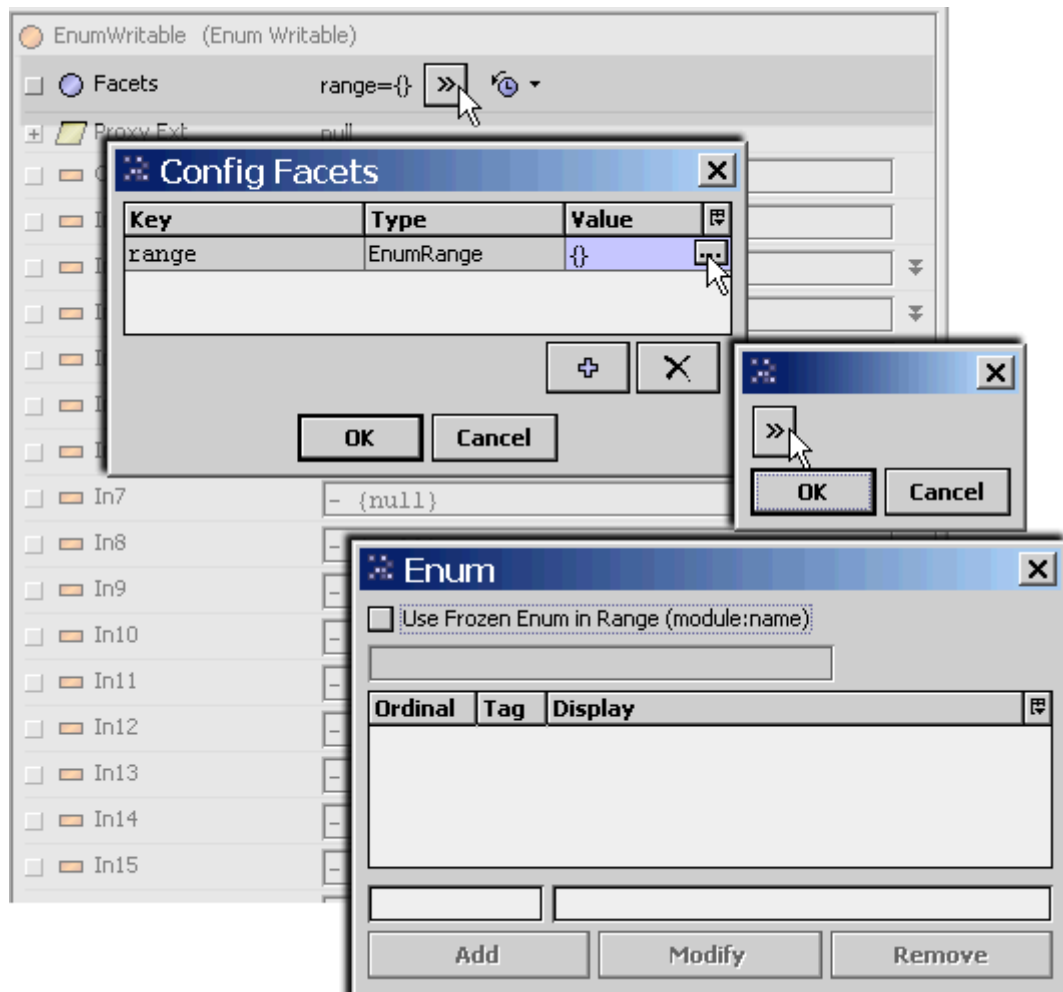
**注意:** 对于 string 类型的 points (StringPoint、StringWritable), facets 通常很少有实际应用。默认情况下, Facets slot 对于 string 类型的 point 是空的。

### Facets 对枚举点 (enum point) 的重要性

Enum (枚举) 类型的组件 (EnumPoint、EnumWritable、EnumSchedule 等) 的 facets 定义了组件的操作范围, 也就是其可能出现的不同枚举状态。每个状态都是通过成对的整数值-文本 (integer value-to-text) 对来定义的, 这种配对也称为 ordinal-tag (序号-标签)。每个序号 (ordinal) 必须是一个唯一的整数, 并且每个标签 (tag) 必须是唯一文本。默认情况下, point 的值使用标签 (tag) 文本进行显示。

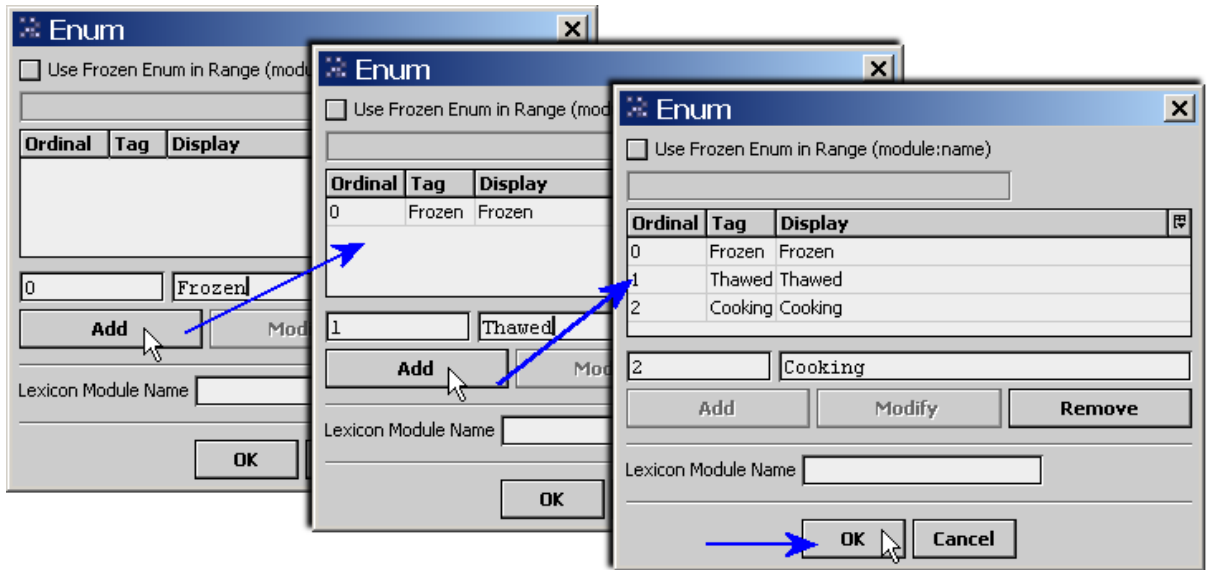
如果从 control 调色板添加了一个 enum point，此时，其 Facets slot 的 range 项是空。直到对该 facet 进行了编辑，并提供一个 ordinal-tag 值，它才能显示整数值。如下图所示，当您编辑 range facets 时，会出现一个特殊的 Enum 窗口。

图 80 Enum point 的“range” facets 产生的 Enum 窗口



如下图所示，在 Enum 窗口中，添加某个条目时，在左侧的 Ordinal 框中，输入一个整数值后，Add 按钮就会被激活。在右侧的 Tag 框中输入相关文本，然后点击 Add，facets 的 range 就被添加上了。完成之后点击 OK，并在剩下的弹出窗口中点击 OK。

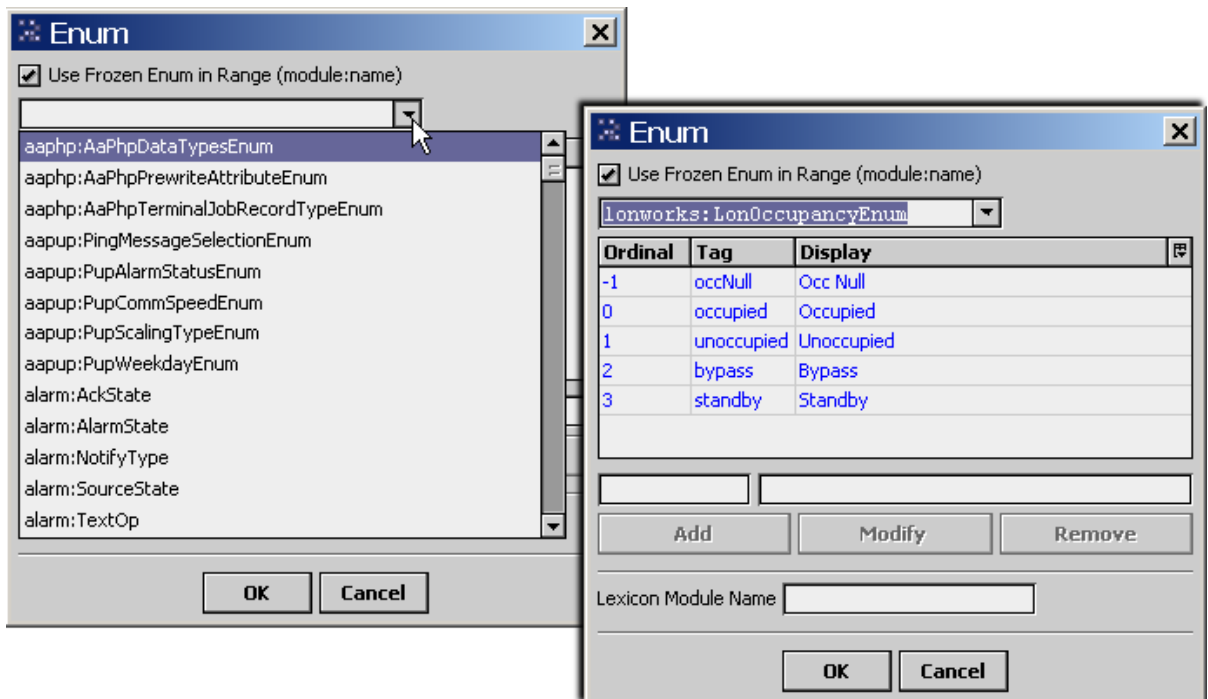
图 81 在 Ordinal 框中输入唯一整数值，并在 Tag 框中输入相关文本



如果使用 lexicon，并且如果 Tag 字符串与 lexicon 文件中的 lexicon“keys”匹配，那么可以在“LexiconModule Name”文本域中输入已配置的模块名（例如，control 或 kitControl）。在这种情况下，枚举将会显示这些序数（ordinals）对应的 lexicon 字符串（值），而不是 tag 文本。

在定义枚举的 range 时，除了像上面那样进行自定义外，也可以从众所周知的 Frozen 枚举中进行选择，这些选项都是在各种已安装的模块中定义的。此处有个复选框让您可以实现这一点，它提供了一个下拉列表，供您按照模块和枚举类型进行选择。

图 82 Enum 窗口中的 frozen enum 选择



根据 driver/network 类型，当添加 enum 类型的代理点时，设备（device）下的 PointManager 可能会自动配置该 facets 的 range。例如，在一个 Lonworks 设备（device）下，如果您添加了一个 EnumPoint 到 Lonworks NVO，这个 LonworksNVO 使用了枚举的 SNVT，那么这个 point 的 facets 将自动被配置正确的 range 值。

**注意：**如果一个 enum 类型的 point 收到一个其定义的 facets 的 range 内没有的 input 值，那么它将会显示该 input 的 ordinal 整数值。这与 r2 Niagara 中所使用的多状态对象不同，在 r2 Niagara 中，对于“stateText”条目中没有定义的任何值，将会显示“Error”。

### Facets 对点动作 (Point actions) 的影响

对于某些带有 actions 的 Points 来说（请参考[点动作](#)，第 82 页），facets 还会影响 point 的 action（命令）菜单的可用性。

- EnumWritable

当收到一个 override 或 emergency 命令时，辅助下拉选择会显示用显示的 tag 文本列出可能的枚举值（在它的 range 中）。该列表将会按 ordinal 从低到高的顺序从上到下显示与之相关的 tag

- NumericWritable

当收到一个 override 或 emergency 命令时，输入窗口只允许输入在 facets 的“min”和“max”值之间的某个数值，包括“min”和“max”值。默认情况下，numeric 类型的 points 的这些 facets 值为 min = -inf, max = +inf（这时就不会检查 action 的有效范围）。

例如，使用 NumericWritable 的该 facet 的功能，通过将 facets 设为 min= 65 和 max=85，可以设置一个温度控制设定。保存该修改后，任何对该 NumericWritable 的 override 或 emergency 动作都必须在该范围内。不然，用户会看到一个有关可接受输入数据范围的消息，并且会被提示重新操作。

**注意：**Facets 的“min”和“max”值不影响接收到的任何 input 值或代理数据，而只影响通过 action 产生的值。

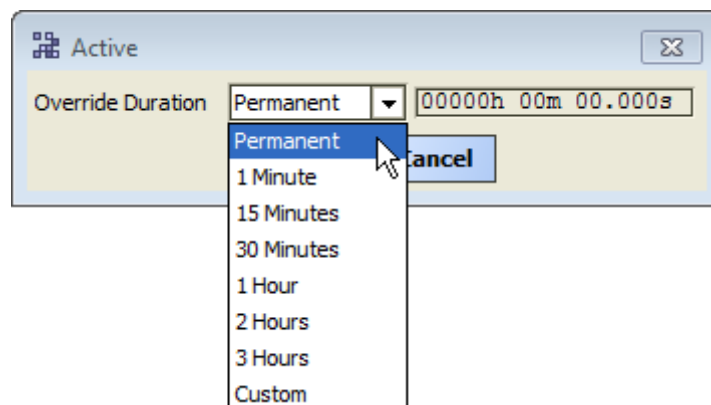
- Maximum override duration（针对任何可写类型的点）

使用 facets，还可以限制在可 writable control point 上触发的手动（第 8 优先级）override 动作的 Maximum override duration。默认情况下，writable point 的手动覆盖（override）没有时长限制。更多详细信息，请参考[“Maximum override duration facet”](#)，第 89 页。

### Maximum override duration facet

有时可用（但是之前的文档没有提过），可以限制 control point 上触发的 action 的最大覆盖时间。默认情况下，writable point 的手动（第 8 优先级）override 在持续时间方面没有限制，因此在 action 菜单中的默认值是“Permanent”。

图 83 Writable control point 的默认 override 动作菜单

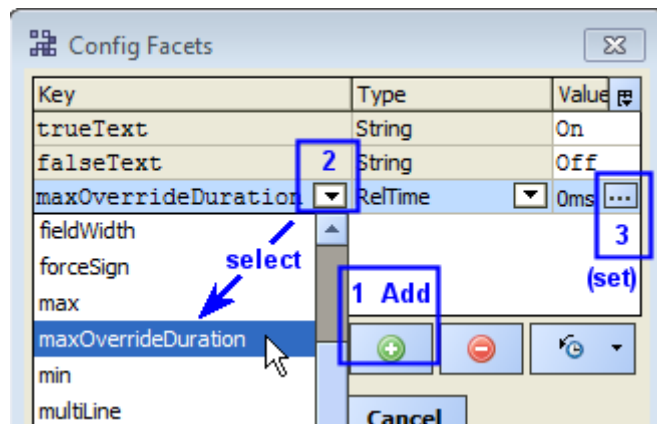


如果需要，在下面任何一种对象中或在二者之上都添加一个带指定持续时间的“maxOverrideDuration”facet（选择 baja:RelTime 类型）可以改变上述情况：

- Config-Sys Info 属性
- 任何 writable control point

**注意：**Override 只限制操作员 override（第 8 优先级）的影响，因此 emergency override（第 1 优先级）在持续时间方面没有限制。换句话说，一个 emergency 级别的 override 会一直持续到执行 emergency 级别的“auto”。

图 84 选择 maxOverrideDuration 时的 Config Facets 编辑器



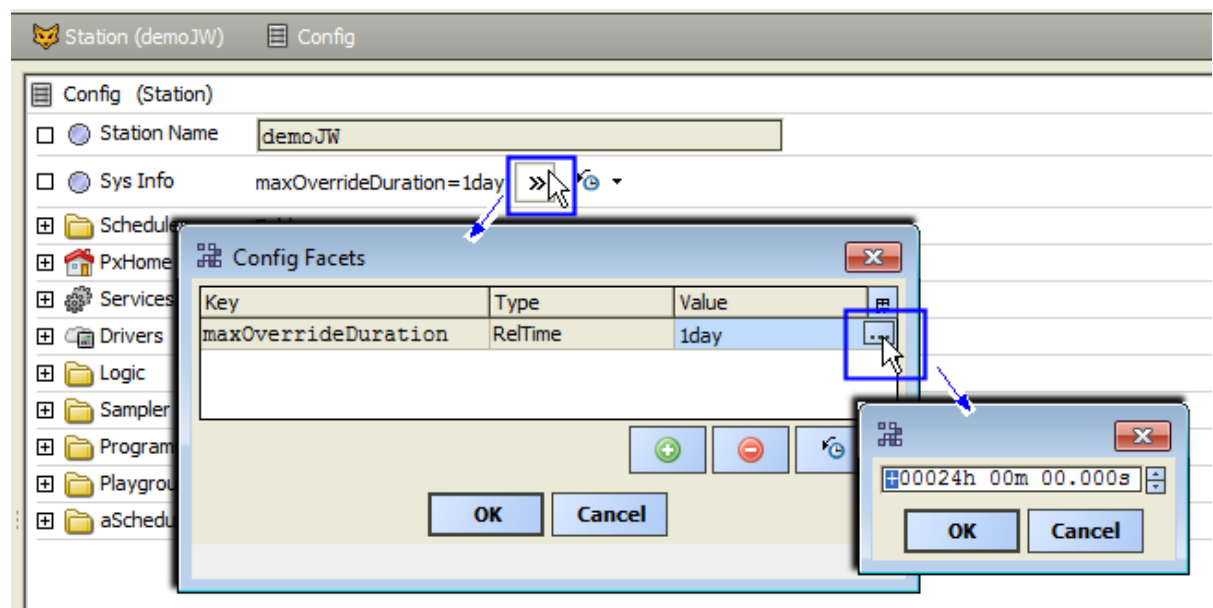
当 writable point 被 maxOverrideDuration 这一 facet 限制时，其 action 菜单也会进行相应的调节，以显示允许的范围。

有关编辑 facets 的详细情况，请参考“访问和编辑 facets”，第 87 页。

### Config - Sys Info 属性

Station 的根组件 Config 的 Sys Info 属性具有 facets 控制，这一属性在 Config 组件的 property sheet 中能看到。

图 85 添加全局 maxOverrideDurationfacet 到 station 的 Config 组件的 SysInfo 属性中

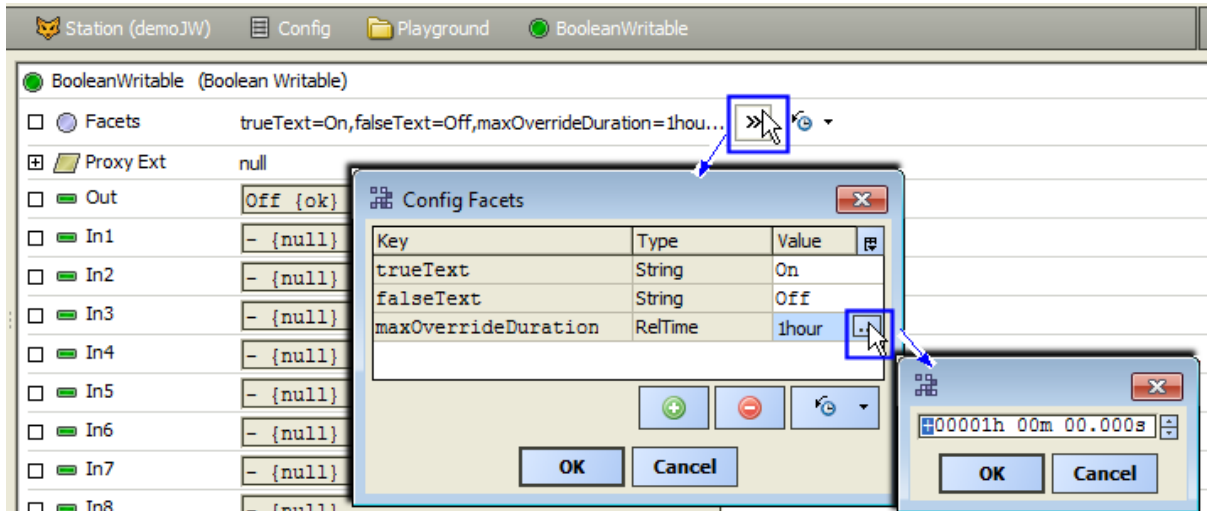


在 Sys Info 属性上添加这一 facet，会全局限制所有没有自己的“maxOverrideDuration”facet 的控制点的手动 overrideaction。

### 任何可写控制点（writable control point）

Station 中的每个 writable control point 可以有单独指定的最大覆盖时长（maximunoverride duration）。如果该 facet 存在，它将会覆盖任何全局的（Sys Info）maxOverrideDuration 值。

图 86 在 point 中添加的 maxOverrideDuration facet（将会覆盖全局设置）

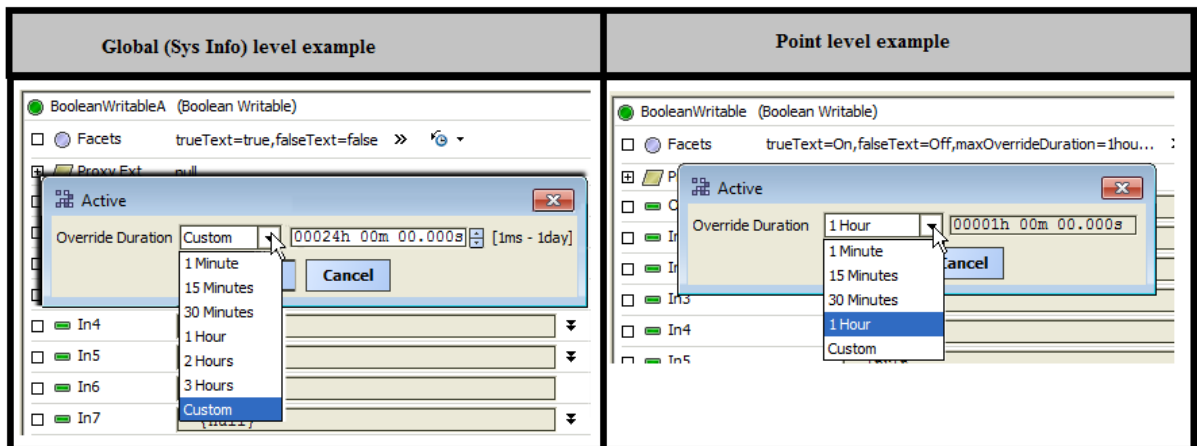


如图所示，该 maxOverrideDurationfacet 可以与 control point 的使用的任何其它 facets 一起添加。上述 BooleanWritable point 的例子已经配置了 trueText 和 falseText 这两种 facets。

### Action 菜单示例

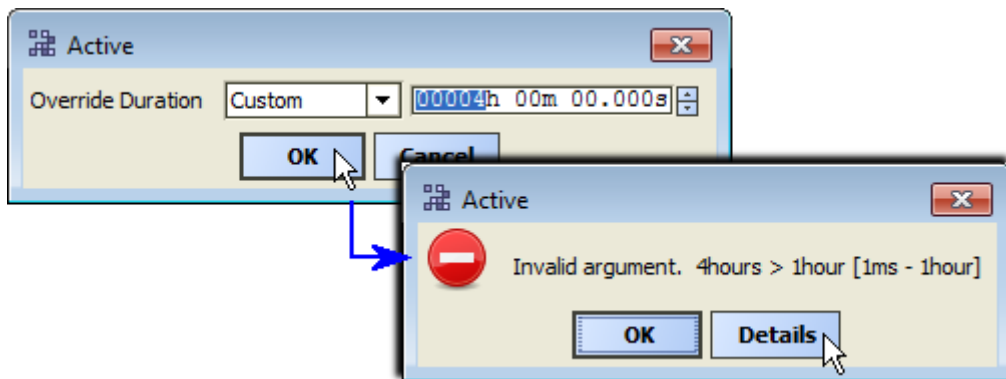
在下面的 action menu 例子中，writablepoint 有 maxOverrideDuration facet。

图 87 受 maxOverrideDurationfacet 影响的 override action 菜单的例子



请注意，如果系统用户试图触发一个超过 maxOverrideDuration 指定限值的自定义 override，将会弹出一个如下图所示的显示了覆盖时长范围的窗口。

图 88 自定义 Override 尝试超过 maxOverrideDuration 限值会生成错误弹出窗口



如上图所示，会在 [方括号]显示允许的时长范围，在本例中为[1ms - 1hour]。