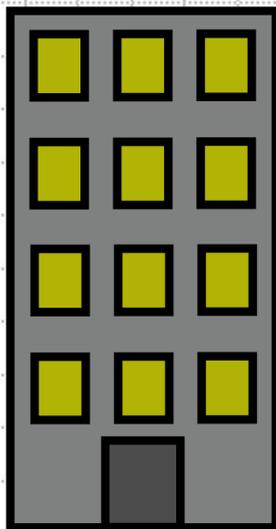


# BACnet Lighting 2011



New BACnet objects and services have been proposed to support lighting control.

What can you do with them?

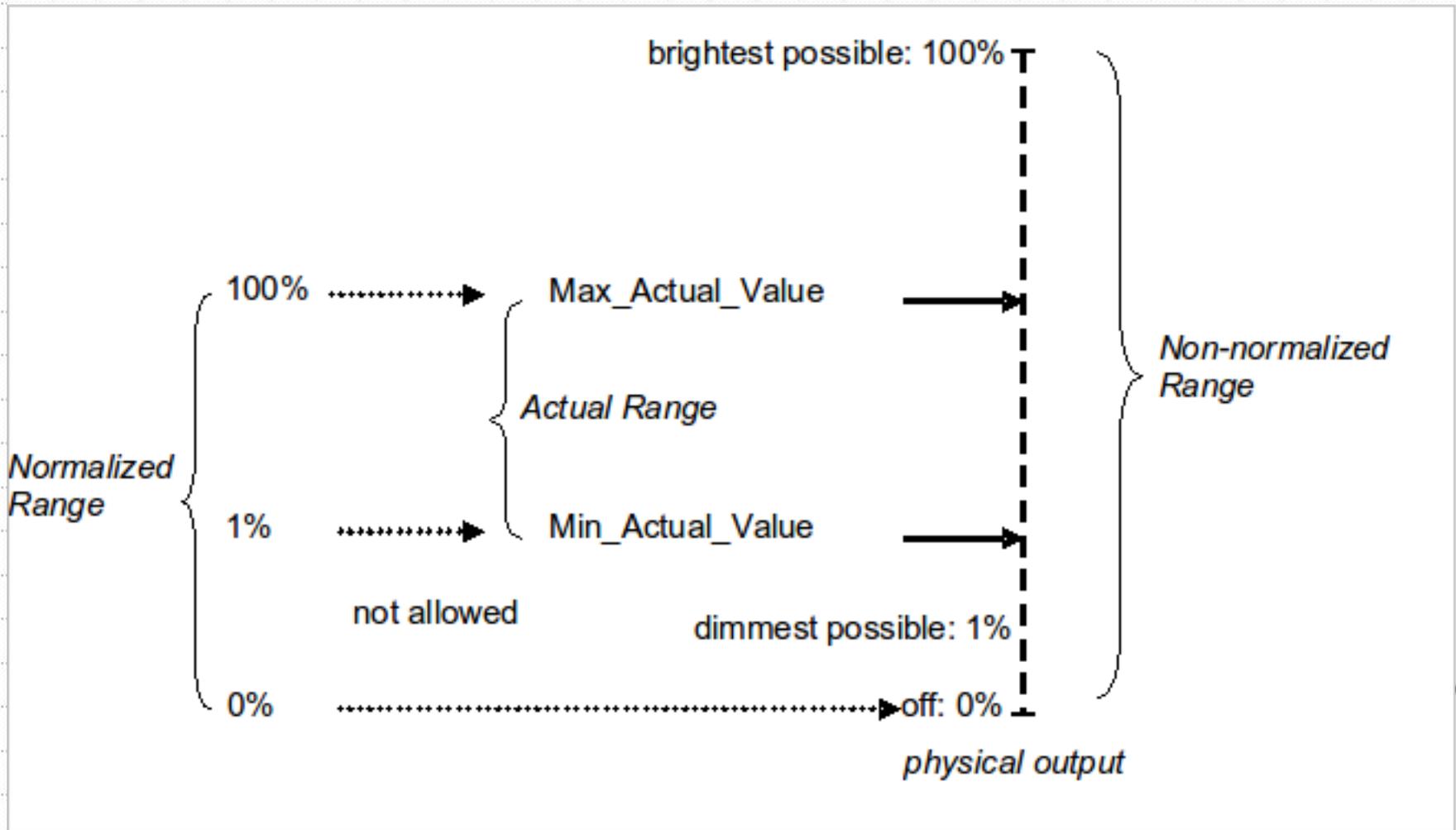
What can you do to prepare your devices?

# Lighting Output Object

- Present\_Value in Percent of Normalized Range
  - 0.0%=Off
  - 1.0-100% =On
  - -1.0 = Warn
  - -2.0 = Warn Relinquish
  - -3.0 = Warn Off
- Lighting\_Command for dimming control
  - operation, target level, ramp rate, fade time, step, priority
  - Operations include:
    - fade-to, ramp-to
    - step-up, step-down, step-off
    - warn, warn-relinquish, warn-off
    - stop
- Internal dimming engine drives Tracking\_Value
- Internal Egress\_Time timer for automatic relinquish

# Normalized Range

**Figure 12-X1.** Normalized Range of the Lighting Output



# Linear Interpolation

```
float linear_interpolate(  
    float x1,  
    float x2,  
    float x3,  
    float y1,  
    float y3)  
{  
    float y2;  
  
    if (y3 > y1) {  
        y2 = y1 + (((x2 - x1) * (y3 - y1)) / (x3 - x1));  
    } else {  
        y2 = y1 - (((x2 - x1) * (y1 - y3)) / (x3 - x1));  
    }  
  
    return y2;  
}
```

# Normalize Percent to Steps

```
uint16_t dimmer_percent_to_step(  
    uint8_t percent)  
{  
    uint16_t steps = 0;  
  
    if (percent) {  
        steps = linear_interpolate_int(  
            1, percent, 100,  
            1, DIMMER_STEP_MAX);  
    }  
  
    return steps;  
}
```

# Scale Between Min and Max

```
uint16_t dimmer_level_scale_steps(  
    uint8_t level,  
    uint8_t low_limit, uint8_t high_limit)  
{  
    uint16_t steps = 0, low_limit_steps = 0, high_limit_steps = 0;  
  
    if (level) {  
        low_limit_steps = linear_interpolate_int(  
            1, low_limit, 100,  
            1, DIMMER_STEP_MAX);  
        high_limit_steps = linear_interpolate_int(  
            1, high_limit, 100,  
            1, DIMMER_STEP_MAX);  
        steps = linear_interpolate_int(  
            1, level, 100,  
            low_limit_steps, high_limit_steps);  
    }  
  
    return steps;  
}
```

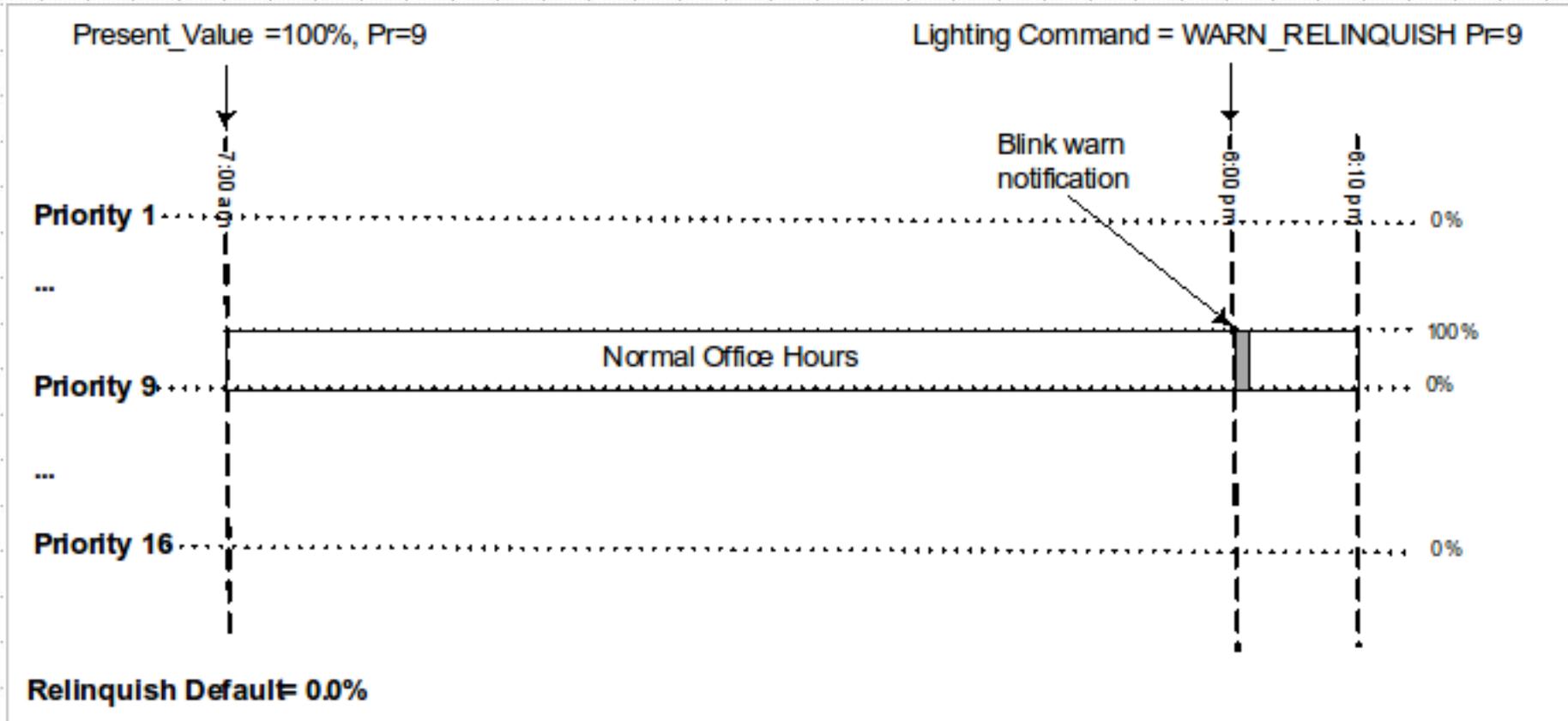
# Lighting Commands

**Table 12-X2.** Special Values of the Present\_Value Property

Value	Description
-1.0	WARN
-2.0	WARN_RELINQUISH
-3.0	WARN_OFF

# Warn-Relinquish

Figure 12-X2. Daily Schedule with Blink-Warn Example



# Lighting Command Operations

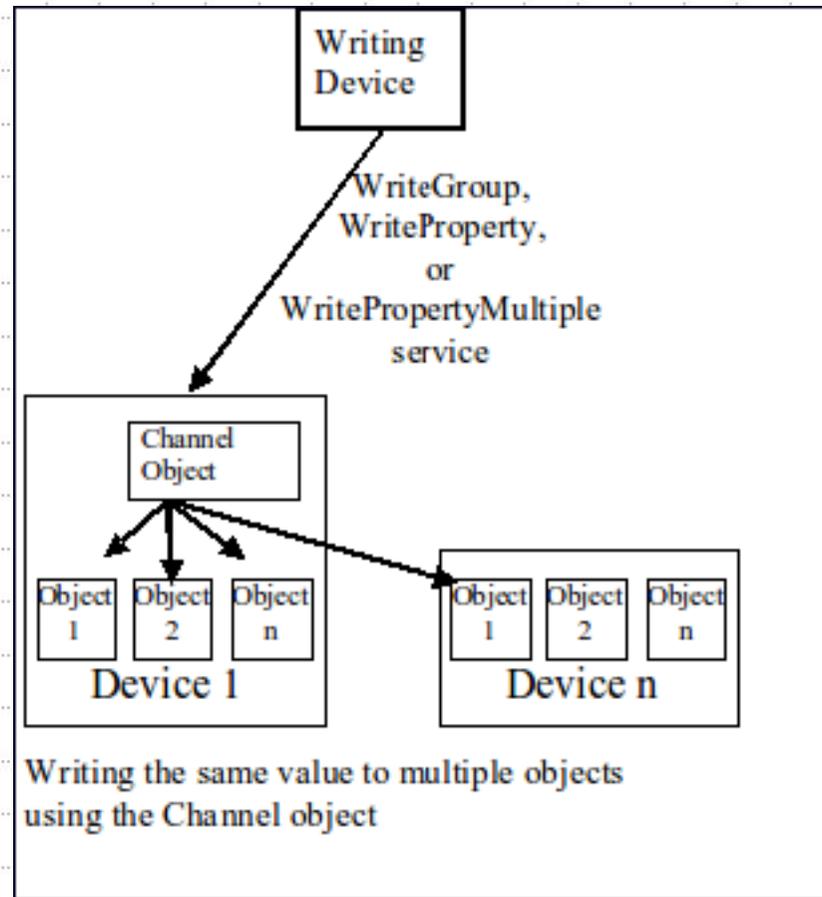
- Parameters:
  - operation
  - target-level
  - ramp-rate
  - fade-time
  - step-increment
  - priority
- fade-to
- ramp-to
- step-up, step-down, step-off
- warn, warn-relinquish, warn-off
- stop

# Channel Object

- Present\_Value is Channel Value (pseudo Commandable)
  - similar to Any - primitives
  - includes Lighting Command
- Propagated value to List of Object Property References
  - Object Property Reference can be another device
  - Execution Delay + Inhibit
  - Channel Number
  - Group membership
- Defined data coercion

# Write Inside or Outside Objects

**Figure 12-X1.** Channel object behavior



# WriteGroup Service

- Broadcast Write to Channel Objects in a Single Group
- Group number is meant to limit
  - 0=reserved
  - 1=min
  - 4,294,967,295=max
- Priority - used if not specified
- Change List - Tuples
  - Channel Number = property of Channel object 1-65535
  - Overriding Priority - specific priority if needed
  - Value - may be coerced if not exact datatype
- Inhibit Delay - if allowed by Channel object

# Write Group to Channel object

**Figure 12-X2.** Control Groups limit WriteGroup effect to specific Channel objects across many devices

