Most of the robots and schedule entries in this document do not utilize any Wink App robots or Schedules, they are designed to operate independently of the Wink App other than using the devices specified in each flow. When putting a device name into the flow it must match the case, spelling, and spacing EXACTLY as in the Wink App.

Red text should be edited to fit your system
Blue text are optional changes
Grey may not apply to every system and can be removed
Green is information about a specific line

Schedule Entries

Basic Schedule Entry

At 6:40pm turn the island light on and the living room group and set to 75%

```
if(hours==18 & minutes==40) // After dinner
{
   node.send(context.global.executeWinkCMD("Island","light","on","75"));
   node.send(context.global.executeWinkCMD("Living Room","group","on","75"));
   node.send(WinkCMDmsg);
   send_ui_note('information',10*60*1000,'Island and Living Room at 75% via
schedule',Math.floor(Math.random()*1000));
}
```

Fade In Schedule Entry

At 6:30am on weekdays fade the bedroom lamp from 0 to 100 over a 900 second period

```
if(hours==6 & minutes==30 && intday!==0 && intday!=6)
{
   effect="fadein";
   o_name="Bedroom";
   o_type="light";
   min=0;
   max=100;
   period=900;
   WinkCMDmsg = context.global.executeEffectCMD(effect,o_name,o_type,min,max,period);
   node.send(WinkCMDmsg);
   send_ui_note('information',10*60*1000,'Good Morning Fade In',Math.floor(Math.random()*1000));
}
```

Note: intday!==0 && intday!=6 means not Sunday or Saturday

Sunrise based Schedule Entry

Lights on at Dawn

```
if(hours==(context.global.sunTimes.goldenHour.hour) &&
minutes==context.global.sunTimes.goldenHour.minute) // Indoor Lights at Dawn
{
   node.send(context.global.executeWinkCMD("Living Room","group","on","100"));
   node.send(context.global.executeWinkCMD("Kitchen","group","on","100"));
   node.send(WinkCMDmsg);
   send_ui_note('information',10*60*1000,'Kitchen and Living Room on via
schedule',Math.floor(Math.random()*1000));

}
```

Turns lights off an hour after Sunrise

```
if(hours==(context.global.Weather.SunriseHour+1) && minutes==context.global.Weather.SunriseMin)
// Indoor Lights off after sunrise
{
   node.send(context.global.executeWinkCMD("Living Room","group","off","0"));
   node.send(context.global.executeWinkCMD("Kitchen","group","off","0"));
   node.send(WinkCMDmsg);
   send_ui_note('information',10*60*1000,'Kitchen and Living Room off via
schedule',Math.floor(Math.random()*1000));
}
```

Robots

Outlink on at certain times when door opens

Times aren't easily used in robot tab so I have created variables in the schedules tab that define certain times.

```
//--------------Time based Contexts-----------
// 545 to 615 is context.global.earlyMorning true/false
// 1700 to 2100 is context.global.evening true/false
// 2100 to 0000 is context.global.lateEvening true/false
// 2300 to 0500 is context.global.overnight true/false
```

The time is defined in the schedules tab then the variable is called on in the robots tab this example shows my earlyMorning variable but it could be named any unique name

```
if(typeof context.global.earlyMorning=="undefined")
{
   context.global.earlyMorning=false;
}
if ((context.global.earlyMorning===false) && ((hours==5 && minutes>=45) || (hours==6 &&
minutes<=45)))
{
   context.global.earlyMorning=true;
}
if((context.global.earlyMorning===true) && (((hours>=6) && (hours<=23)) && ((hours>=0) &&
(hours<=5))))
{
   context.global.earlyMorning=false;
}
if(context.global.DEBUG){ node.warn(context.global.earlyMorning); } // Debug conditional
```

Then this is in the robots tab

```
if ((context.global.earlyMorning=="true") && (changed.name=="Front Door" && changed.old_state!
=="Opened" && changed.new_state=="Opened" ))
{
   try {
      node.send(context.global.executeWinkCMD("Master Bath","light","on","100"));
      node.send(WinkCMDmsg);
      node.send(context.global.send_ui_note('information',600000,'Early Morning Bath Heat
On',Math.floor(Math.random()*1000)));
   }
   catch(error){
```

```
        node.warn(error.message);
}}
```

I use a delay timer to turn the "Master Bath" outlink off after 30 minutes

```
if (changed.name=="Master Bath" && changed.old_state.powered=="Off" &&
changed.new_state.powered=="On")
{
    setTimeout(function(){
    try {
        node.send(context.global.executeWinkCMD("Master Bath","light","off","0"));
        node.send(WinkCMDmsg);
        node.send(context.global.send_ui_note('information',600000,'Master Bath
off',Math.floor(Math.random()*1000)));
    }
    catch(error){
        node.warn(error.message);
    }
},30*60*1000 );
}
```

The line },30*60*1000); is defining the delay by },minutes x seconds x milliseconds;

Light or group off when another light is turned off

This robot turns all kitchen lights off when the island is turned on and left on for 15 minutes between 9pm and 5am. Use case would be someone forgetting to turn the light off.

```
//                                  Kitchen off with island at night
if ((changed.name=="Island" && changed.old_state.powered=="Off" &&
changed.new_state.powered=="On") && ((context.global.lateEvening===true)||
(context.global.overnight===true)))
{
    setTimeout(function(){
    if(context.global.winkState.light_bulbs.Island.powered===true)
     try {
            node.send(context.global.executeWinkCMD("Kitchen","group","off","0"));
             node.send(WinkCMDmsg);
            node.send(context.global.send_ui_note('information',300000,'Island left
on',Math.floor(Math.random()*1000)));
        }
        catch(error){
            node.warn(error.message);
        }
    } ,15*60*1000 );
}
```

Light on when another light is turned on

```
//                                  Cabinet on with Island evening
if ((changed.name=="Island" && changed.old_state.powered=="Off" &&
changed.new_state.powered=="On") && ((context.global.evening=="true")||
(context.global.lateEvening===true))
    try {
        node.send(context.global.executeWinkCMD("Cabinet","group","on","100"));
        node.send(WinkCMDmsg);
        node.send(context.global.send_ui_note('information',300000,'Cabinet on with
Island',Math.floor(Math.random()*1000)));
    }
    catch(error){
        node.warn(error.message);
    }
```

Light on when a door opens then have it fade back to previous state

Thanks Ken Vermillion

Anything following"//" is a note for information about the line

```
//                                        Light on when front door opens at night
if ((changed.name=="Front Door" && changed.new_state=="Opened") &&
((context.global.overnight===true )||(context.global.lateEvening===true))) //check if front door opens
{
   if (context.global.winkState.light_bulbs['Center Can'].powered===true)
   {
      var lrlamp = (context.global.winkState.light_bulbs['Center Can'].brightness)*100;
 // sets variable to current brightness of Center Can

      node.send(context.global.executeWinkCMD("Center Can","light","on","50"));
// now turn light to 50% brightness
      node.send(WinkCMDmsg);

      var timerId = setTimeout(function()
      {node.send(context.global.executeWinkCMD("Center Can","light","on",lrlamp));
 // turn same light back to previous brightness after 45 seconds
      node.send(WinkCMDmsg)},45000);
   }
   else
   {
      node.send(context.global.executeWinkCMD("Center Can","light","on","50"));
// now turn light to 50% brightness
      node.send(WinkCMDmsg);

      var timerId = setTimeout(function()
      {node.send(context.global.executeWinkCMD("Center Can","light","off"));
// turn same light back to previous brightness after 45 seconds
      node.send(WinkCMDmsg)},45000);
   }
}
```

Lock door during overnight if closed for 5 mins

```
if ((changed.name=="Main Door" && changed.old_state!=="Closed" && changed.new_state=="Closed" )
&& (context.global.overnight===true))
 {
    setTimeout(function(){

      try {
         // This command locks main  door if closed for 5 mins during overnight
         WinkCMDmsg = context.global.executeWinkCMD("Entry Lock","lock","lock");
         node.send(WinkCMDmsg);
         send_ui_note('information',30*60*1000,'Overnight locking front
door',Math.floor(Math.random()*1000));
      }

       catch(error){
             node.warn(message);
          }
  },300000);
  }
```

Turn on a light using a tripper

```
if (changed.name=="Attic" && changed.old_state!=="Opened" && changed.new_state=="Opened" )
  {
      try {
         // This command turns my Hallway light on to 100%
         WinkCMDmsg = context.global.executeWinkCMD("Attic Light","light","On","100");
         node.send(WinkCMDmsg);
      }
      catch(error){
         node.warn(error.message);
      }
  }
```

Turn off a light using a tripper

```
if (changed.name=="Attic" && changed.old_state!=="Closed" && changed.new_state=="Closed" )
  {
     try {
        // This command turns my Hallway light on to 100%
        WinkCMDmsg = context.global.executeWinkCMD("Attic Light","light","Off","0");
        node.send(WinkCMDmsg);
     }
     catch(error){
        node.warn(error.message);
     }
  }
```

Advanced Schedules and Robots

Shuts Leaksmart valve if sensors detect a leak

```
if ((changed.name=='Laundry Water Sensor' || changed.name=='Kitchen Water Sensor' ||
changed.name=='Bathroom Water Sensor') && (changed.old_state!==true &&
changed.new_state===true))
        try {
            // This command shuts water off in house if a leak is detected
            WinkCMDmsg = context.global.executeWinkCMD("Water Shut Off","valve","close");
            node.send(WinkCMDmsg);
            pmsg=context.global.sendViaPushBullet('note',"Water leak detected by " + changed.name + "
main valve closed",'Water leak in house');
            node.send(pmsg);

        }
        catch(error){
            node.warn(error.message);
        }
```

Turn on lights when Ring Doorbell detects motion and turn them off again 10 minutes after motion stops. Also will only run if the house is empty (no presence)

```
if ((changed.name=='Doorbell' && changed.old_state!==true && changed.new_state===true) && (!
context.global.checkPresence()))
        try {
            // This command turns on certsain lights when the door detects motion
            WinkCMDmsg = context.global.executeWinkCMD("Couch lamp","light","on","100");
            WinkCMDmsg = context.global.executeWinkCMD("Sabra lamp","light","on","100");
            WinkCMDmsg = context.global.executeWinkCMD("Hallway Light","light","on","100");
      WinkCMDmsg = context.global.executeWinkCMD("Living Room Celing","group","on","100");
            node.send(WinkCMDmsg);
            pmsg=context.global.sendViaPushBullet('note','Motion detect lights on','Who goes there');
            node.send(pmsg);
        }
        catch(error){
            node.warn(error.message);
        }
```

```
if ((changed.name=='Doorbell' && changed.old_state!==false && changed.new_state===false) && (!
context.global.checkPresence()))
   {
      var timer = setTimeout(function()
         {
            if (context.global.winkState.sensor_pods['Doorbell'].motion===false)              {


               try {
                  WinkCMDmsg = context.global.executeWinkCMD("Couch lamp","light","off","0");
                  WinkCMDmsg = context.global.executeWinkCMD("Sabra lamp","light","off","0");
                  WinkCMDmsg = context.global.executeWinkCMD("Hallway Light","light","off","0");
                  WinkCMDmsg = context.global.executeWinkCMD("Living Room Ceiling","group","off","0");
                  node.send(WinkCMDmsg);
                  pmsg=context.global.sendViaPushBullet('note','No motion lights off','all quiet');
                  node.send(pmsg);
      }
      catch(error){
         node.warn(error.message);
      }
}
         },600000);
   }
```

Turn humidifier on and off using Spotter or any other device that reports humidity levels

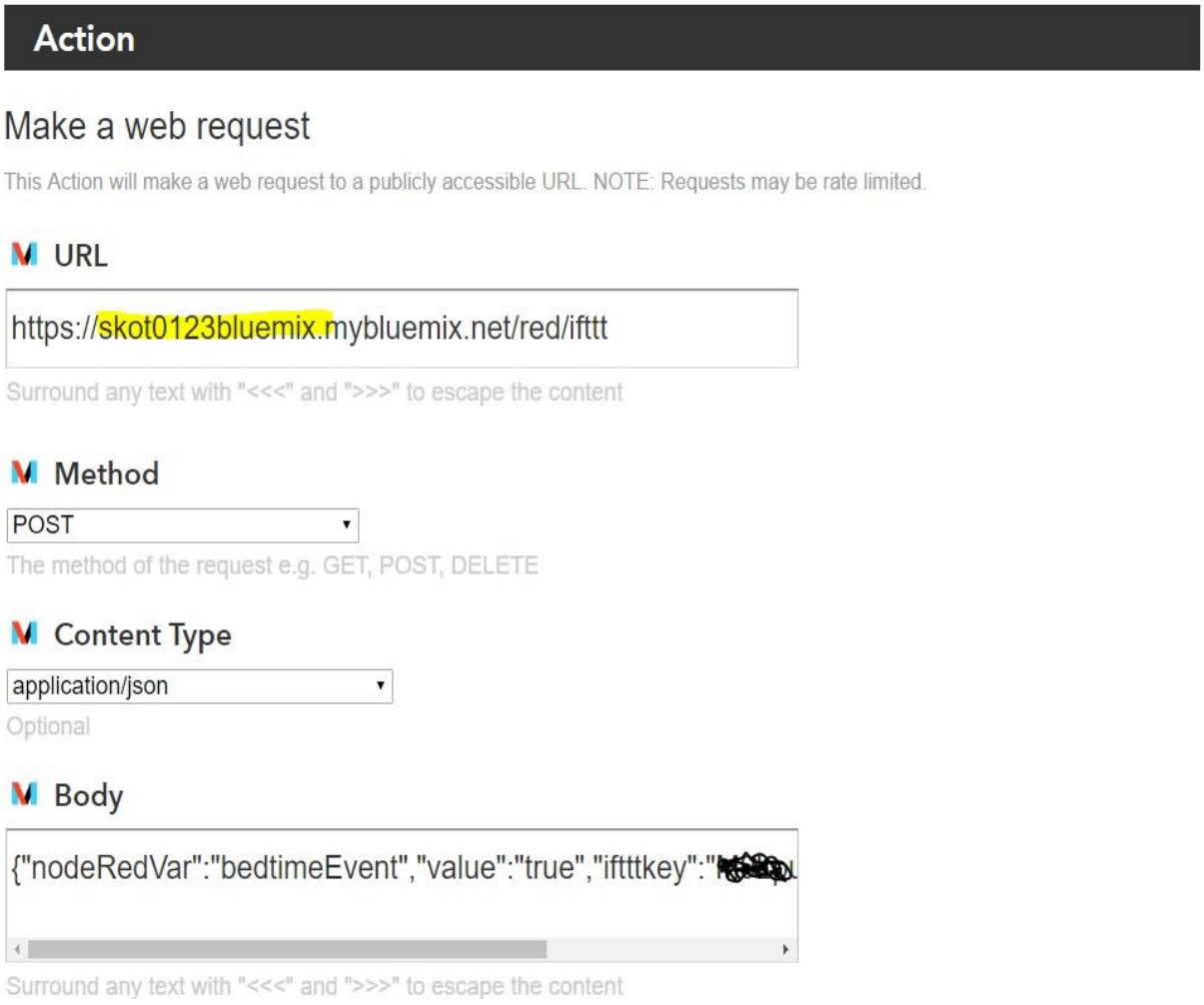
```
if (typeof context.global.highHumidity=="undefined")
{
   context.global.highHumidity=0;
}

if(context.global.winkState.sensor_pods['Master Bedroom'].humidity<=0.34 &&
context.global.highHumidity===0)
{
          try {
          WinkCMDmsg = context.global.executeWinkCMD("Humidifier","light","on","100");
          node.send(WinkCMDmsg);
          send_ui_note('information',30*60*1000,'Low Humidity
Detected',Math.floor(Math.random()*1000));

           context.global.highHumidity=1;
}
```

```
            catch(error){
                node.warn(message);
            }
}


if(context.global.winkState.sensor_pods['Master Bedroom'].humidity>0.35 &&
context.global.highHumidity===1)
{
        try{
        WinkCMDmsg = context.global.executeWinkCMD("Humidifier","light","off","0");
        node.send(WinkCMDmsg);
        send_ui_note('information',30*60*1000,'High Humidity
Detected',Math.floor(Math.random()*1000));

        context.global.highHumidity=0;
}
catch(error){
            node.warn(message);
        }

}
```

Using Echo via IFTTT to set variables and trigger WNR actions

Create IFTTT recipe with Echo custom Phrase as If then Maker as below changing highlighted to your url

See https://github.com/tfatykhov/WinkRedNode/blob/master/README-IFTTT.md for help with IFTTT integration



The schedule entry below is triggered by the above Echo/IFTTT event and turns off the Kitchen Group, Bedroom Group, Left Lamp, and sets the Right Lamp to 1%. It also sends a message to the activity feed and sets my alarm variable to 1 (arms alarm).

Red text should be edited to fit your system
Blue text is optional changes
Grey may not apply to every system and can be removed

It is important to make sure that your device names match the Wink names exactly in case, spelling, and spacing

```
if(context.global.bedtimeEvent=="true")
try {
   node.send(context.global.executeWinkCMD("Kitchen","group","off","0"));
   node.send(context.global.executeWinkCMD("Bedroom","group","off","0"));
   node.send(context.global.executeWinkCMD("Left lamp","light","off","0"));
   node.send(context.global.executeWinkCMD("Right lamp","light","on","1"));
   node.send(WinkCMDmsg);
   node.send(context.global.send_ui_note('information',300000,'House is set for bedtime
mode',Math.floor(Math.random()*1000)));
   context.global.alarmArmed=1;
   context.global.bedtimeEvent=false;

}
   catch(error){
   node.warn(error.message);
}
```

When you say "trigger bedtime" ,or whatever phrase, the variable is made true and the above fires then changes variable back to false.

Bloomsky Integrated Lights

This schedule entry turns lights off during the day if the Bloomsky reads a luminance of above 3350 between one hour after sunrise and one hour before sunset and someone is home and on if the reading is below 3350

```
if(hours > context.global.Weather.SunriseHour+1 && hours < context.global.Weather.SunsetHour-1 &&
context.global.checkPresence())
{
   setTimeout(function()
   {
      if(context.global.Weather.Bloomsky.Luminance<3350 && context.global.winkState.light_bulbs['Left
```

```
lamp'].powered===false)
      {
         WinkCMDmsg=context.global.executeWinkCMD("Livingroom","group","on","100");
         node.send(WinkCMDmsg);
         send_ui_note('information',30*60*1000,'Cloudy day lights turning
on',Math.floor(Math.random()*1000));
      }
      else if(context.global.Weather.Bloomsky.Luminance>=3350 &&
context.global.winkState.light_bulbs['Left lamp'].powered===true)
      {
         WinkCMDmsg=context.global.executeWinkCMD("Livingroom","group","off","0");
         node.send(WinkCMDmsg);
         send_ui_note('information',30*60*1000,'Sunny day lights turning
off',Math.floor(Math.random()*1000));
      }
   },10*60*1000);
}
```

Change Nest Thermostat depending on an individual's presence

```
// if Angie presence is yes and after 8am and before 5pm and temp outside is over 77
degrees and Angie home during day isn't already running.

if(context.global.Presence.Angie.home=="no")
{
   context.global.AngieHome=false;
   context.global.AngieHomeDuringDay=0;
}

if(context.global.Presence.Angie.home==="undefined")
{
   context.global.AngieHome=false;
}if((context.global.AngieHome===true) && (hours>=8) && (hours<=17) &&
(context.global.Weather.Bloomsky.TemperatureF>=77) && context.global.AngieHomeDuringDay!=1)
{
   node.send(context.global.executeTstatCMD('Home Home Thermostat','cool_start_at','23.5'));
   node.send(WinkCMDmsg);
   send_ui_note('information',300*60*1000,'A/C set to 74',Math.floor(Math.random()*1000));
   context.global.AngieHomeDuringDay=1;
}

if(context.global.Presence.Angie.home=="yes")
{
   context.global.AngieHome=true;
}
```

Start of PushBullet Notification flows. These go in robot tab.

Basic entry to send PushBullet Notification in Robots

```
pmsg=context.global.sendViaPushBullet('note','Header','Message');
node.send(pmsg);
```

Lower Cabinet Opened

This flow notifies me if any of my lower cabinets are opened for more than 5 secs. Helps me know if my little girl is getting into stuff that she shouldn't.

```
 if ((changed.name=="Kitchen Sink" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Pantry" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Corner" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Pots And Pans" && changed.old_state!=="Opened"
&& changed.new_state=="Opened") || (changed.name=="Bread" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Can Goods" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Baking" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Crock Pot" && changed.old_state!=="Opened" &&
changed.new_state=="Opened"))
   {
      setTimeout(function()
         {
               if ((changed.name=="Kitchen Sink" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Pantry" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Corner" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Pots And Pans" && changed.old_state!=="Opened"
&& changed.new_state=="Opened") || (changed.name=="Bread" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Can Goods" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Baking" && changed.old_state!=="Opened" &&
changed.new_state=="Opened") || (changed.name=="Crock Pot" && changed.old_state!=="Opened" &&
changed.new_state=="Opened"))
            {
               try {
                  pmsg=context.global.sendViaPushBullet('note', changed.name + ' Cabinet
Opened','Where is Nikki');
                  node.send(pmsg);
               }
               catch(error){
                  node.warn(error.message);
               }
            }
         },5000);
   }
```

Door Locked

This flow notifies me when my front door is locked

```
if (changed.name=="Entry Lock" && changed.old_state!=="Locked" && changed.new_state=="Locked")
try{
 pmsg=context.global.sendViaPushBullet('note','Front Door Locked','House is locked');
node.send(pmsg);
}
catch(error){
node.warn(error.message);
}
```

Door Unlocked

This flow notifies me when my front door unlocks

```
 if (changed.name=="Entry Lock" && changed.old_state!=="Unlocked" &&
changed.new_state=="Unlocked")
try{
 pmsg=context.global.sendViaPushBullet('note','Front Door Unlocked','House is unlocked');
node.send(pmsg);
}
catch(error){
node.warn(error.message);
}
```

Mail is Here

This flow notifies me when my mailbox is opened

```
 if (changed.name=="Mailbox" && changed.old_state!=="Opened" && changed.new_state=="Opened")
try{
 pmsg=context.global.sendViaPushBullet('note','Mail is here','Mailbox opened');
node.send(pmsg);
}
catch(error){
node.warn(error.message);
}
```

Propane is Low

This flow notifies me when my propane levels get low on my barbcue.

```
 if (changed.name=="Grill Tank" && changed.old_state>=".2" && changed.new_state<".2")
try{
 pmsg=context.global.sendViaPushBullet('note','Propane is low','Replace propane tank');
node.send(pmsg);
}
catch(error){
node.warn(error.message);
}
```

Fridge Left Open

This flow notifies me if my Fridge door was left open for 5 mins.

```
 if (changed.name=="Fridge" && changed.old_state=="Closed" && changed.new_state=="Opened")
{
    setTimeout(function(){
        if(changed.name=="Fridge" && changed.old_state=="Closed" && changed.new_state=="Opened")
    try{
 pmsg=context.global.sendViaPushBullet('note','Fridge left open','Close the refrigerator');
node.send(pmsg);
}
catch(error){
node.warn(error.message);
}
} ,300000);
}
```

Freezer Left Open

This flow notifies me if my Freezer door was left open for 5 mins.

```
 if (changed.name=="Freezer" && changed.old_state=="Closed" && changed.new_state=="Opened")
{
   setTimeout(function(){
      if(changed.name=="Freezer" && changed.old_state=="Closed" && changed.new_state=="Opened")
   try{
 pmsg=context.global.sendViaPushBullet('note','Freezer left open','Close the freezer');
node.send(pmsg);
}
catch(error){
node.warn(error.message);
}
} ,300000);
}
```

Presence Based Robots

To check for someone's presence in an if statement use context.global.checkPresence()

Day Presence

```
if(typeof context.global.nopresence=="undefined")
{
   context.global.nopresence=0;
}
//                                    No Presence Day
if(context.global.daylight==1 && !context.global.checkPresence() && context.global.nopresence==1)
{
   node.send(context.global.executeWinkCMD("All Lights","group","off","0"));
   node.send(context.global.executeTstatCMD('Home Home Thermostat','users_away','true'));
   node.send(WinkCMDmsg);
   node.send(context.global.send_ui_note('information',300*60*1000,'No one is home, setting system to
away mode',Math.floor(Math.random()*1000)));
   pmsg=context.global.sendViaPushBullet('note','No Presence','The house has been set to away mode');
   node.send(pmsg);
   context.global.nopresence=0;
}
```

```
//                                          Presence Day
if(context.global.daylight==1 && context.global.checkPresence() && context.global.nopresence===0)
{
   node.send(context.global.executeTstatCMD('Home Home Thermostat','users_away','false'));
   node.send(WinkCMDmsg);
   node.send(context.global.send_ui_note('information',300*60*1000,'Welcome
Home',Math.floor(Math.random()*1000)));
   pmsg=context.global.sendViaPushBullet('note','Presence','Welcome Home, house set for presence');
   node.send(pmsg);
   context.global.nopresence=1;
}
```

Night Presence

```
//                                          No Presence Night
if(context.global.daylight===0 && !context.global.checkPresence() && context.global.nopresence==1)
{
   node.send(context.global.executeWinkCMD("Kitchen","group","off","0"));
   node.send(context.global.executeWinkCMD("Bedroom","group","off","0"));
   node.send(context.global.executeWinkCMD("Right Lamp","light","on","20"));
   node.send(context.global.executeWinkCMD("Left Lamp","light","off","0"));
   node.send(context.global.executeTstatCMD('Home Home Thermostat','users_away','true'));
   node.send(WinkCMDmsg);
   send_ui_note('information',300*60*1000,'No one is home, setting house to away
mode',Math.floor(Math.random()*1000));
   pmsg=context.global.sendViaPushBullet('note','No Presence','The house has been set to away mode');
   node.send(pmsg);
   context.global.nopresence=0;
}
//                                      Presence Night
if(context.global.daylight===0 && context.global.checkPresence() && context.global.nopresence===0)
{
   node.send(context.global.executeTstatCMD('Home Home Thermostat','users_away','false'));
   node.send(context.global.executeWinkCMD("Left Lamp","light","on","50"));
   node.send(context.global.executeWinkCMD("Right Lamp","light","on","50"));
   node.send(WinkCMDmsg);
   node.send(context.global.send_ui_note('information',300*60*1000,'Welcome
Home',Math.floor(Math.random()*1000)));
   pmsg=context.global.sendViaPushBullet('note','Presence','Welcome Home, house set for presence');
   node.send(pmsg);
   context.global.nopresence=1;
}
```

This robot activates the wink PRESENCE shortcut if Ken arrives and Katie was already home

Thanks Ken

```
if(changed.name=="Ken" && changed.old_state===false && changed.new_state===true)
  // check if Ken's presence is now detected
{
        if(context.global.winkState.sensor_pods.Katie.presence===true)
 // check if Katie is already home
        {

        node.warn("you've made it past the katie home kenny arrive check");
// This is a test message in activity feed that verifies that the above has been completed
        try {
        WinkCMDmsg = context.global.executeWinkCMD("PRESENCE","shortcut");
// activate PRESENCE shortcut
        node.send(WinkCMDmsg);
        }
        catch(error){
        node.warn(error.message);
        }
        }
        }
```

Alarm Robots

Alarm checks if door was opened by a recent arrival

This alarm checks for a family member's arrival after sensing door opening. If no arrival within the specified time and failsafe switch is not on the alarm sounds

```
// This part goes in schedule tab

 // testing arrival
var d = new Date();
var t = d.getTime()/1000; //this will calculate current time on the server in your local time zone
var kdiff = Math.trunc((t -context.global.winkState.sensor_pods.Kaitlen_Geo.lastUpdated)/60); // this will
give you difference in minutes.
var adiff = Math.trunc((t -context.global.winkState.sensor_pods.Angie_Geo.lastUpdated)/60); // this will
give you difference in minutes.
var sdiff = Math.trunc((t -context.global.winkState.sensor_pods.Scott_Geo.lastUpdated)/60); // this will
give you difference in minutes.

if(kdiff<=10 && context.global.winkState.sensor_pods.Kaitlen_Geo.lastEvent=="enter" &&
context.global.winkState.sensor_pods.Kaitlen_Geo.lastWaypoint=="home")
{
   context.global.someoneArrived=true;{
setTimeout(function(){
```

```
    context.global.someoneArrived=false;
},6*60*1000 );
}}

if(adiff<=5 && context.global.winkState.sensor_pods.Angie_Geo.lastEvent=="enter" &&
context.global.winkState.sensor_pods.Angie_Geo.lastWaypoint=="home")
{
    context.global.someoneArrived=true;{
setTimeout(function(){

    context.global.someoneArrived=false;
},6*60*1000 );
}}

if(sdiff<=5 && context.global.winkState.sensor_pods.Scott_Geo.lastEvent=="enter" &&
context.global.winkState.sensor_pods.Scott_Geo.lastWaypoint=="home")
{
    context.global.someoneArrived=true;{
setTimeout(function(){

    context.global.someoneArrived=false;
},6*60*1000 );
}}

if(typeof context.global.someoneArrived=="undefined")
{
    context.global.someoneArrived=false;
}

// This is the actual alarm robot

if ((changed.name=="Front Door"||changed.name=="Back Door") && (changed.old_state=="Closed" &&
changed.new_state=="Opened")&& context.global.alarmArmed==1)
{
    setTimeout(function(){
    if((context.global.winkState.light_bulbs.Island.powered===false) &&
(context.global.someoneArrived===false))
     try {
            node.send(context.global.executeWinkCMD('Siren','siren','siren_only','null'));
            node.send(context.global.send_ui_note('information',600000,changed.name +'
Alarm!!!',Math.floor(Math.random()*1000)));
            node.send(WinkCMDmsg);
            pmsg=context.global.sendViaPushBullet('note', changed.name + ' Alarm!!!','alarm has been
triggered');
            node.send(pmsg);
        }
        catch(error){
            node.warn(error.message);
```

```
        }
    } ,30*1000 );
}
```

Front Door Opens with No One Home

This robot detects front door open when no one is home, if the island light is not turned on within 30 seconds the siren is activated and I get a pushbullet notification.

```
if ((changed.name=="Front Door" && changed.old_state=="Closed" && changed.new_state=="Opened")
&& (!context.global.checkPresence()) && context.global.duringDay=="true" )
{
    setTimeout(function(){
    if((context.global.winkState.light_bulbs.Island.powered===false) && (!context.global.checkPresence()))
     try {
            node.send(context.global.executeWinkCMD('Siren','siren','siren_only','null'));
               node.send(WinkCMDmsg);
            node.send(context.global.send_ui_note('information',600000,'Door
Alarm!!!',Math.floor(Math.random()*1000)));
            pmsg=context.global.sendViaPushBullet('note','Alarm!!!','alarm has been triggered by front door');
            node.send(pmsg);
        }
        catch(error){
            node.warn(error.message);
        }
    } ,30*1000 );
}
```

Front Door Opens with Individual Arrival

This robot detects front door open and if alarm is armed using alarmArmed variable==1 it waits 30 seconds then checks to see if Kaitlen just arrived. If she didn't just arrive it activates siren and notifies via pushbullet. Also has the failsafe of Island being turned on to prevent alarm

```
if ((changed.name=="Front Door" && changed.old_state=="Closed" && changed.new_state=="Opened")
&& context.global.alarmArmed==1)
{
    setTimeout(function(){
    if(((context.global.winkState.sensor_pods.Kaitlen.presence===false)) &&
(context.global.winkState.light_bulbs['Island'].powered===false))
     try {
            node.send(context.global.executeWinkCMD('Siren','siren','siren_and_strobe','null'));
node.send(WinkCMDmsg);
            node.send(context.global.send_ui_note('information',600000,'Door
Alarm!!!',Math.floor(Math.random()*1000)));
            pmsg=context.global.sendViaPushBullet('note','Alarm!!!','alarm has been triggered by front door');
            node.send(pmsg);
```

```
        }
        catch(error){
            node.warn(error.message);
        }
    } ,30*1000 );
}
```

Motion Based Robots

Close Garage Door after 30 seconds if no motion

```
if (changed.name=="Garage Door" && changed.old_state=="Closed" && changed.new_state=="Opened")
            {
            node.warn("garage door has been opened");
            var timerGrg = setTimeout(function()
            {if(context.global.winkState.sensor_pods['Garage Motion'].motion===false)
            {WinkCMDmsg=context.global.executeWinkCMD("Close Garage Sequence","shortcut");
            node.send(WinkCMDmsg)}},30000);
            }
```

Kitchen light on

```
if (changed.name=="Kitchen" && changed.old_state!==true && changed.new_state===true)
{
        try {
            // This command turns my kitchen light on to 100%
            WinkCMDmsg = context.global.executeWinkCMD("Kitchen Light","light","On","100");
            node.send(WinkCMDmsg);
        }
        catch(error){
            node.warn(error.message);
        }
    }
```

Kitchen light off after 10 mins no motion

```
 if (changed.name=="Kitchen" && changed.old_state!==false && changed.new_state===false)
{
   var timerBasementMotion = setTimeout(function()
      {
         if(context.global.winkState.sensor_pods['Kitchen'].motion===false)
         {
            try {
               node.send(context.global.executeWinkCMD("Kitchen Light","light","off","0"));
               node.send(context.global.send_ui_note('information',30*60*1000,'No Motion Kitchen Light
off',Math.floor(Math.random()*1000)));
            }
            catch(error){
               node.warn(error.message);
            }
         }
      },600000);
```

Remote Based Robots

Activating a schedule using a remote

```
      if (changed.name=="Master Bedroom Remote" && context.global.winkState.remotes['Master
Bedroom Remote'].button_off_pressed===true) //Robot: if bottom button on MB remote pushed, activate
Alexa Bedtime Shortcut
      {
      node.warn("bedroom button pressed");
      WinkCMDmsg = context.global.executeWinkCMD("Alexa Bedtime","shortcut");
      node.send(WinkCMDmsg);
      }
```

Quick Reference

Quick tips to check CURRENT STATE of Wink devices.

Presence: context.global.winkState.sensor_pods['Person Name'].presence===true

Motion Sensor: context.global.winkState.sensor_pods['Sensor Name'].motion===true

Trippers: context.global.winkState.sensor_pods['Tripper Name'].opened===true OR .closed===true

Switches: context.global.winkState.binary_switches['Switch Name'].powered===true OR false

Bulbs: context.global.winkState.light_bulbs['Bulb Name'].powered===true OR .brightness===.8

Locks: context.global.winkState.locks['Lock Name'].locked===true OR .unlocked

Groups: context.global.winkState.groups['Group Name'].powered.or===true OR .powered.and===true

*powered.or checks to see if any lights in the group are powered or not, powered.and checks to see if ALL lights in the group are powered or not

Can also check state of Cameras, Smoke detectors, using the same format as above. Devices names are case sensitive, true/false, locked/unlocked, opened/closed are always lowercase