

```

////////////////////////////////////// CODE HEADER
//////////////////////////////////////
treenode current = parnode(1);
treenode patient = parnode(2);
//////////////////////////////////////
//////////////////////////////////////

{ //***** PickOption Start *****\

/**Determine whether a patient will leave early \n*/

/**\nEarly Exit Condition: */

int PASS = true;

int staffAllocated = getobjecttype(getrequiredresource(patient,getvarnum(patient,
VAR_ActivityRow),STAFF,1)) == OBJECT_Staff;

int condition = /**/ !staffAllocated && getlabel(patient,"PCI") == 3 && time() -
getcreationtime(patient) > 500/**list:PASS~time() > 100~content(current) > 10~getlabel(patient,
"PCI") == 1~getlabel(patient, "Acuity") <= 3~time() - getvarnum(patient, VAR_LastReleaseTime) >
normal(90,10,1)~bernoulli(50,true,false)~time() - getcreationtime(patient) > normal(90,10,1)*/;

if(condition == false)
    return 0;

if(condition)
{
    /**\n\nEarly Exit Location (where patients will be directed when leaving early): */
    treenode lwbs_obj = /**/PatientExit/**scriptpath:VIEW:/modules/HC/picklists/getlocations*/;
    //set the patient's destination to the
    set(getvarnode(patient, VAR_PatientDestination), tonum(lwbs_obj));
    //mark patient as LWBS
    set(getvarnode(patient, VAR_LWBS), 1);

    /**\n\nSet the patient's label named */
    string label_name = /**/"a_LWBS"*/;
    /** to the value */
    double label_value = /**/1/**/;
    /** to identify the patient as an early exit.*/

    if(stringlen(label_name) > 1)
    {
        //create label if necessary, and set its value
        set(assertlabel(patient, label_name, DATATYPE_NUMBER), label_value);
    }

    //connect to first location within the early exit area (if not already)
    int connected = 0;
    for(int curport = 1; curport <= nrop(current) && connected == 0; curport++)
    {
        if(outobject(current, curport) == lwbs_obj)
            connected = 1;
    }
    if(!connected)
    {
        contextdragconnection(current, lwbs_obj, "A");
        openallip(lwbs_obj);
    }

    terminateactivity(patient, getvarnum(patient,VAR_ActivityRow));

    // Change the patient's current activity's type to "Patient Travels Unattended" (regardless
of what activity type
    // it is supposed to be), so that the patient will walk out rather than possibly be escorted
out.
    int curActivityRow = getvarnum(patient, VAR_ActivityRow);
    settablenum(getvarnode(patient, VAR_ActivityTable), curActivityRow, COL_ActivityType,
ACTIVITY_PatientTravelsUnattended);
}
}

```

```
    //release patient so they can travel to exit now
    releaseitem(patient, findmatch(nrop(current), outobject(current, count) == lwbs_obj));
}

/**\n\nOn every evaluation event, each patient in the queue will first have the Early Exit
Condition evaluated for them. If the patient passes the condition (returns true), then the patient
will leave the queue immediately and walk to the specified early exit location.*/

} //***** PickOption End *****\
```