

```

// Kill script, based on a script from Void Singer
// All credit goes to Void Singer
// and all blame for bugs to Jenni Darkwatch for messing it up. :)
//
// How to use:
// 1. Rez (or wear) object you want to scrub
// 2. Drop this script into the root prim
// 3. Follow instructions :)

integer gIntChn;

KillScripts(){
    integer vIntCnt = llGetInventoryNumber( INVENTORY_SCRIPT );
    // llOwnerSay((string)vIntCnt+" scripts in link "+(string)llGetLinkNumber());
    string vStrInv;
    integer i;
    for(i=0;i<vIntCnt;i++){
        vStrInv=llGetInventoryName( INVENTORY_SCRIPT, i);
        if (llGetScriptName()!=vStrInv && vStrInv!=""){
            llOwnerSay("Link "+(string)llGetLinkNumber()+": Removed "+vStrInv);
            llRemoveInventory(vStrInv);
        }
    }

    llOwnerSay("Link "+(string)llGetLinkNumber()+": Had "+(string)vIntCnt+" scripts, now
    "+(string)(llGetInventoryNumber( INVENTORY_SCRIPT )-1)+" scripts.");
    llRemoveInventory( llGetScriptName() );
}

default{
    state_entry(){
        if (2 > llGetLinkNumber()){
            integer vIntCnt = llGetNumberOfPrims();
            while (1 < vIntCnt){
                llGiveInventory( llGetLinkKey( vIntCnt-- ), llGetScriptName() );
            }
            llOwnerSay( "Take me to Inventory and ReRez me to finish the process." );
        }else{
            KillScripts();
        }
    }

    on_rez( integer vInt ){
        if (2 > llGetLinkNumber()){
            llOwnerSay( "Please select this object, then go to the Build menu and 'Set Scripts to
Running in Selection' to finish removing the scripts." );
            llSetScriptState( llGetScriptName(), FALSE );
            llSleep( 2.0 );
            KillScripts();
        }
    }
}

```

}

}