

```

function evaltF() {
    var f = document.tF;
    var idx = 0;
    var subnet_mask_val = getCfg("lan_mask" + idx).split(".");
    var mask = ["255", "255", "255", subnet_mask_val[3]];
    var lanIP = combinIP2(f.lan_ip);
    var lan_mask = f.subnet_mask.value;
    var intLanMask = parseInt(subnet_mask_val[3]);
    var ulStartIP, ulEndIP, ulNewLanIP, ulNewLanMask;
    var uLlan_ID, uLlan_Broadcast;
    var DhcpStartIP, DhcpEndIP;
    var i, j = 0;
    var found = 0;
    var orig_lanip = getCfg("lan_ip" + idx);
    var orig_mask = getCfg("lan_mask" + idx);
    var orig_dhcp_startip = getCfg("dhcp_startip" + idx);
    var orig_dhcp_endip = getCfg("dhcp_endip" + idx);
    var ipbyte1 = getCfg("lan_ip0").split(".");
    var ipbyte2 = lanIP.split(".");

    var lan_ip = [f.lan_ip[0].value, f.lan_ip[1].value, f.lan_ip[2].value,
f.lan_ip[3].value];

    var lan_brocast = BrocastLong(lanIP, lan_mask);

    var ulMaskedStartIP, ulMaskedEndIP;

    DhcpStartIP = combinIP2(f.dhcp_startip);
    DhcpEndIP = combinIP2(f.dhcp_endip);

    ulStartIP = IpToLong(DhcpStartIP);

    ulEndIP = IpToLong(DhcpEndIP);

    var dhcpS = DhcpStartIP.split(".");
    var dhcpE = DhcpEndIP.split(".");

    dhcp_ip2 = IP2long(dhcpE[0], dhcpE[1], dhcpE[2], dhcpE[3]);
    dhcp_ip1 = IP2long(dhcpS[0], dhcpS[1], dhcpS[2], dhcpS[3]);

    var wait_time = 5;

    DhcpStartIP = f.lan_ip[0].value + "." + f.lan_ip[1].value + "." +
f.lan_ip[2].value + "." + f.dhcp_startip[3].value;
    DhcpEndIP = f.lan_ip[0].value + "." + f.lan_ip[1].value + "." +
f.lan_ip[2].value + "." + f.dhcp_endip[3].value;
    setCfg("domain_name" + idx, getFieldValue(f.domain_name));
    setCfg("dhcp_startip" + idx, DhcpStartIP);

```

```

setCfg("dhcp_endip" + idx, DhcpEndIP);
setCfg("lan_ip" + idx, lanIP);
setCfg("lan_mask" + idx, f.subnet_mask.value);
setCfg("dhcp_enable" + idx,
document.getElementsByName("dhcp_enable")[0].checked ? "1" : "0");
if (document.getElementsByName("v6_dhcp_enable")[0].checked == true) {
    setCfg("IPv6_LAN_Stateless", "1");
    setCfg("IPv6_LAN_DHCP_Mode", "server");
    setCfg("IPv6_LAN_RADVD", "1");
} else {
    setCfg("IPv6_LAN_Stateless", "0");
    setCfg("IPv6_LAN_DHCP_Mode", "");
    setCfg("IPv6_LAN_RADVD", "0");
}
if (orig_lanip != lanIP || orig_mask != lan_mask || orig_dhcp_startip !=
DhcpStartIP || orig_dhcp_endip != DhcpEndIP) {
    ConfirmOSP(arc.i18n._t("advconfig.network.dhcp.label.change_ip"),
        function() {
            wait_time = 60;
            top.G_IP = lanIP;
            top.G_URL = window.location.protocol + "://" + lanIP + "/login.htm";
            var data = {
                "htttoken": ArcBase._t()
            };
            $.ajax({
                method: "POST",
                url: "change_dhcp.cgi",
                data: data,
                dataType: "text",
                timeout: 5000
            });
            return subForm({
                frm: f,
                cmd: subFormSYSBootAndKillPCListDaemon,
                cmdparam: "DHCPRangeChanged",
                wait: wait_time
            });
        },
        function() {
            setCfg("lan_ip" + idx, orig_lanip);
            setCfg("lan_mask" + idx, orig_mask);
            setCfg("dhcp_startip" + idx, orig_dhcp_startip);
            setCfg("dhcp_endip" + idx, orig_dhcp_endip);
            return false;
        });
} else if (document.getElementsByName("dhcp_enable")[1].checked == true ||
document.getElementsByName("v6_dhcp_enable")[1].checked) {
    return subForm({
        frm: f,
        cmd: subFormLANConf,
        wait: wait_time
    });
} else {
    ConfirmOSP(arc.i18n._t("advconfig.network.dhcp.label.change_ip_addressing"))

```

```
,  
function() {  
    return subForm({  
        frm: f,  
        cmd: subFormLANConf,  
        wait: wait_time  
    });  
},  
function() {  
    return false;  
});  
}  
}
```