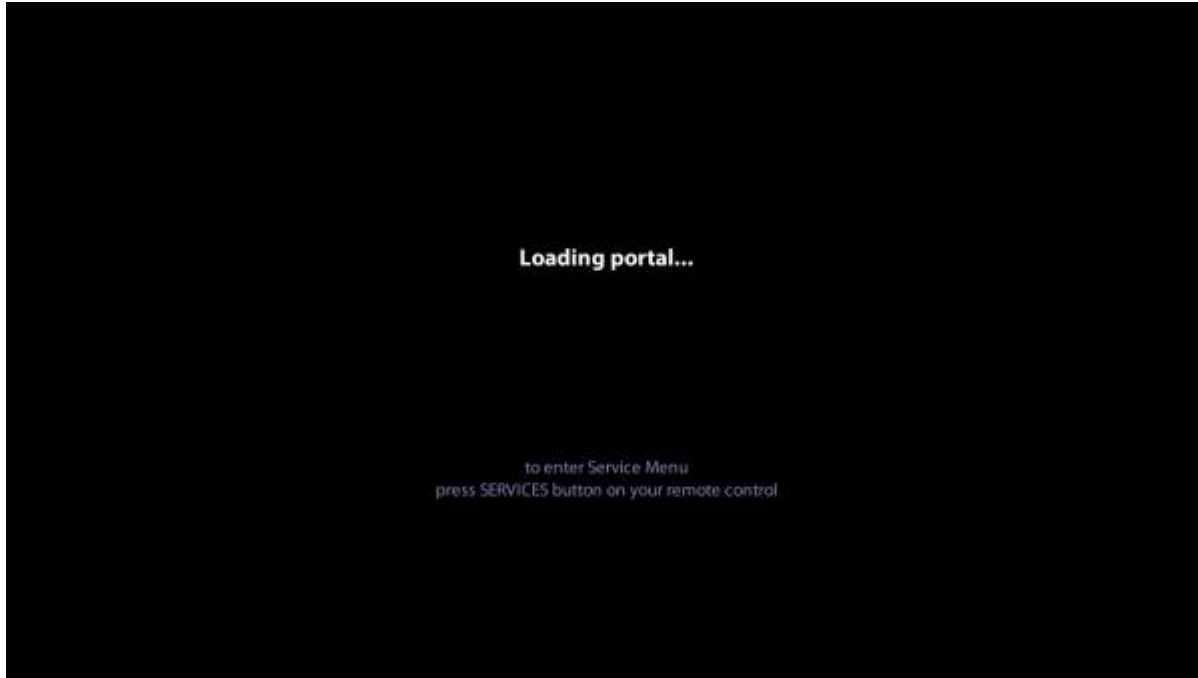


Portals page and embedded portal ("SET/SETUP") disabling while loading

If the values **portal1** and **portal2** have been setup, then there is opportunity to choose them while loading:

<spoiler|Pic.1>



</spoiler>

<spoiler|Pic.2>



</spoiler>

Edit file: `/home/web/index.html`:

- **String 84** - There is black screen while loading, but there is opportunity to open embedded portal by pressing **SET/SETUP**.

```
var BLACK_SCREEN_WHILE_LOADING = 0;
```

change to:

```
var BLACK_SCREEN_WHILE_LOADING = 1;
```

- **String 464** - disabling opportunity to open embedded portal by pressing **SET/SETUP**

```
case 120: // "Services"
    if(!servicePressed){
        infoButtonPressed();
    }
    break;
}
```

change to:

```
case 120: // "Services"
    if(!servicePressed){
        if(!BLACK_SCREEN_WHILE_LOADING){
            infoButtonPressed();
        }
    }
    break;
}
```

- **String 464** - Opening **"System settings"** by pressing **SET/SETUP** button instead of embedded portal opening.

```
case 120: // "Services"
    if(!servicePressed){
        infoButtonPressed();
    }
    break;
}
```

Change to:

```
case 120: // "Services"
    if(!servicePressed){
        gSTB.StartLocalCfg();
    }
    break;
}
```

<spoiler|Example of edit file index.html>

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link rel="stylesheet" type="text/css" title="CSS Stylesheet"
href="style.css">
<title></title>
<script language="JavaScript" src="var_index.js"></script>
<script language="JavaScript" src="loader.js" defer="true"></script>

<style type="text/css">
body {margin:0px}
.ArialBold18{
  font-family: "Myriad Pro";
  font-size:25px;
  font-weight:bold;
  text-align:center;
  color:#FFF;
}
.Verdana14Grey{
  font-family: "Myriad Pro";
  font-size:19px;
  font-weight:normal;
  text-align:center;
  color:#888;
  text-shadow:#0000FF 0 0 20px;
}
#portalsMenu{
  position:relative;
  top:0px;
  width:620px;
  height:150px;
  margin:auto;
  text-align:center;
  display:none;
}
#title{
  position:relative;
  top:0px;
  left:0px;
  width:620px;
  height:50px;
  text-align:center;
  font-family: "Myriad Pro";
  font-size:25px;
  font-weight:normal;
  text-align:center;
  color:#CCC;
}
.menu_table{
  position:relative;
  top:0px;
```

```
width:620px;
height:150px;
margin:auto;
text-align: center;
}
.menu_normal{
height:40px;
font-family:"Myriad Pro";
font-size:30px;
color:#a7b7d6;
}
.fadeBg{
position:absolute;
left:0px;
width:620px;
height:150px;
font-family:"Myriad Pro";
font-size:40px;
color:#FFFFFF;
text-align:center;
line-height: 150px;
margin-top: 0px;
background: url(img/fade_bg.png) no-repeat;
z-index:1;
visibility:hidden;
}
#menu0 {top: -5px}
#menu1 {top: 35px}
</style>

<script>
var BLACK_SCREEN_WHILE_LOADING = 0; // opening portal - black screen
without messages
var rowsTotal = 0, portal_1, portal_2, curPageId, timerRedirect,
timerDhcpPortal, timerToMenu, repeatTimer,
noPortalsURL, servicePressed = false;
var curMenuIdx = null,
repeatTimeout = false, // it is necessary to reset timer for
portals opening
LOADING = '',
PORTAL_LOADING = '',
SERVICE_LOADING = '',
DHCP_PORTAL_LOADING = '',
sm_DirectionMsg = '',
sm_Message = '',
SERVICE_PRESS_INTERVAL = 10000,
CONTINUE = true,
patIP = /[\\n\\w\\S\\s]*PORTAL_IP="(\\S+)*"[\\n\\w\\S\\s]*/,
patPortal1 = /[\\n\\w\\S\\s]*PORTAL_1="(\\S+)*"[\\n\\w\\S\\s]*/,
```

```
patPortal2          = /[\\n\\w\\S\\s]*PORTAL_2="(\\S+)*"[\\n\\w\\S\\s]*/;

var PORTAL_NAME_MAX_LENGTH = 24,
    CUT_STRING_SYMBOL      = '...';

function init(){
  if (!STB_EMULATION) {
    initXpcom();
  }
  try{
    stb.InitPlayer();
  }catch(e){
  }
  //alert(document.location.search);
  window.resizeTo(720, 576);
  x=(screen.width - 720)/2
  y=(screen.height - 576)/2
  window.moveTo(x, y);
  if(/nms/i.test(window.location.search)){
    _debug('NMS version');
    BLACK_SCREEN_WHILE_LOADING = 1;
  }
  _debug('***** INDEX.html : STB STARTED HERE *****');
  curLangIdx = getCurrentLanguage();
  // _debug('curLang = '+getCurrentLanguage());
  stb.EnableServiceButton(false);
  loadScript ('lang/'+curLangIdx+'/resource.js', 'fillPage()');
  failTimer = setTimeout(languageResourcesFailed, NO_LANGUAGE_TIMEOUT);
}
/*function ifNoLanguage(){                                     //function
of language setting if there is wrong values
  setEnvironmentValue('language','en');
  setTimeout(curLangIdx = getCurrentLanguage(),2000)
  _debug('curLang = '+getCurrentLanguage());
  loadScript ('lang/'+curLangIdx+'/resource.js', 'fillPage()');
}
*/
function fillPage(){
  if (!checkLanguageResourceFile()) {
    return;
  }
  if (!BLACK_SCREEN_WHILE_LOADING){
    LOADING = '<span class="ArialBold18">' + pmenu_Loading + '</span>';
    PORTAL_LOADING = '<span class="ArialBold18">' + pmenu_PortalLoading +
'</span>';
    DHCP_PORTAL_LOADING = '<span class="ArialBold18">' +
pmenu_DhcpPortalLoading + '</span>';
    SERVICE_LOADING = '<span class="ArialBold18">' +
pmenu_ServiceMenuLoading + '</span>';
    //document.body.style.background = 'url(new_menu/img/576/bg.jpg) no-
```

```
repeat';
  _debug("background SETUP!")
}
// Delaying main init() for background
setTimeout(init1,200);
}

function init1(){
  var a = check_portals();

  switch (a) {
    case 0:
      noPortalsURL = "services.html"; // open service menu by default
      sm_DirectionMsg = SERVICE_LOADING;
      sm_Message = '';
      var time = 1;
      var use_portal_dhcp =
getEnvironmentValue('use_portal_dhcp').toString();
      if (use_portal_dhcp == "true"){
        var portal_dhcp = readFromStb_URL ('portal_dhcp', '');
        if (portal_dhcp){
          // переходим на портал DHCP
          noPortalsURL = portal_dhcp;
          sm_DirectionMsg = DHCP_PORTAL_LOADING;
          sm_Message = pmenu_PressServiceButton;
          time = SERVICE_PRESS_INTERVAL;
        }
      }
      if (BLACK_SCREEN_WHILE_LOADING){
        document.location = noPortalsURL;
        return;
      }
      else{
        document.getElementById('menu_container').innerHTML =
sm_DirectionMsg;
        document.getElementById('gotoServ').innerHTML = sm_Message;
        timerDhcpPortal = setTimeout(redirectNoPortals,time);
      }
      break;

    case 1:
      var time = 1;
      var use_portal_dhcp =
getEnvironmentValue('use_portal_dhcp').toString();
      if (use_portal_dhcp == "true" &&
!(/nms/i.test(window.location.search))){
        var portal_dhcp = readFromStb_URL ('portal_dhcp', '');
        if (portal_dhcp ){
          // переходим на портал DHCP
```

```
        noPortalsURL = portal_dhcp;
        sm_DirectionMsg = DHCP_PORTAL_LOADING;
        sm_Message = pmenu_PressServiceButton;
        time = SERVICE_PRESS_INTERVAL;
    _debug(noPortalsURL);
    document.location = noPortalsURL;
    timerDhcpPortal = setTimeout(redirectNoPortals,time);
    }
}

// one portal - open it in 3 seconds
if (BLACK_SCREEN_WHILE_LOADING){
    // no messages on the screen, no background image, page div is
display = none
    document.getElementById("pageIndex").style.display = 'none';
}
else{
    // send message to the screen, background should be loaded
    document.getElementById("menu_container").innerHTML =
PORTAL_LOADING;
    document.getElementById('gotoServ').innerHTML =
pmenu_PressServiceButton;
}
timerRedirect = setTimeout(redirect,SERVICE_PRESS_INTERVAL);

break;

case 2:
    var time = 1;
    var use_portal_dhcp =
getEnvironmentValue('use_portal_dhcp').toString();
    if (use_portal_dhcp == "true" &&
!(/nms/i.test(window.location.search))){
        var portal_dhcp = readFromStb_URL ('portal_dhcp', '');
        if (portal_dhcp){
            // open DHCP portal
            noPortalsURL = portal_dhcp;
            sm_DirectionMsg = DHCP_PORTAL_LOADING;
            sm_Message = pmenu_PressServiceButton;
            time = SERVICE_PRESS_INTERVAL;
            _debug(noPortalsURL);
            document.location = noPortalsURL;
            timerDhcpPortal = setTimeout(redirectNoPortals,time);
        }
    }

// both portals are presented - open portals menu
if (BLACK_SCREEN_WHILE_LOADING){
    // background was disabled - turning on it
    //document.body.style.background = 'url(img/main.png) no-repeat';
```

```
        setTimeout (continue_TwoPortals,1);
        return;
    }
    else{
        document.getElementById('menu_container').innerHTML = LOADING;
        sm_Message = pmenu_PressServiceButton;
        timerToMenu = setTimeout(init_continue,1);
    }
    break;
}
rowsTotal = a;
curMenuIdx = 0;
}

function redirectNoPortals(){

    document.location = noPortalsURL;
}

function continue_TwoPortals(){
    rowsTotal = 2;
    curMenuIdx = 0;
    sm_Message = pmenu_PressServiceButton;
    timerToMenu = setTimeout(init_continue,1);
}

function init_continue(){
    document.getElementById('gotoServ').innerHTML = sm_Message;
    if (servicePressed) {
        document.location = "services.html";
        return;
    }
    try{
        stb.SetVideoState(0);
    }catch(e){
        _debug(e)
    }
    show_menu();
}

// Function is checking portal records in CFG and returning their quantity
beginning from 0
function check_portals(){
    var ret = 0;
    if (portal_1 = getPortalName('portal1')) // getPortalName("Portal1")
        ret++;
    if (portal_2 = getPortalName('portal2')) // getPortalName("Portal2")
        ret++;
    return ret;
}
```



```
}

function beforeLoadingPortal(msg){
  if (BLACK_SCREEN_WHILE_LOADING){
    document.body.style.background = 'none';
    document.getElementById("pageIndex").style.display = 'none';
  }
  else{
    document.getElementById("menu_container").innerHTML = msg;
    document.getElementById("gotoServ").innerHTML = "";
  }
}

// There is portal_1 or portal_2
function redirect(){
  var p, url;
  beforeLoadingPortal(PORTAL_LOADING);
  if(!portal_1){
    portal_1 = portal_2;
  }
  p = getProtoAndHostname(portal_1);
  if (p.protocol) {
    url = portal_1;
  }
  else{
    url = 'http://' + portal_1;
  }
  location.href = url;
}

function show_menu(){
  var b = '';
  b += '<div id="portalsMenu">';
  b += '<div id="title"></div>';
  b += '<div class="menu_table">';
  b += '<div class="menu_normal" id="td0"></div>';
  b += '<div class="menu_normal" id="td1"></div>';
  b += '</div>';
  b += '<div class="fadeBg" id="menu0"></div>';
  b += '<div class="fadeBg" id="menu1"></div>';
  b += '</div>';
  //
  document.getElementById('menu_container').innerHTML = b;
  document.getElementById('title').innerHTML = pmenu_PortalChoice;
  document.getElementById('td0').innerHTML =
cutString(portal_1,PORTAL_NAME_MAX_LENGTH);
  document.getElementById('td1').innerHTML =
cutString(portal_2,PORTAL_NAME_MAX_LENGTH);
  document.getElementById('menu0').innerHTML =
cutString(portal_1,PORTAL_NAME_MAX_LENGTH);
  document.getElementById('menu1').innerHTML =
```

```
cutString(portal_2, PORTAL_NAME_MAX_LENGTH);
    document.getElementById("portalsMenu").style.display = "block";
    menuItem_Select(curMenuIdx);
}

function cutString(str, len){
    var a = str;
    if (a.length > len){
        a = str.substr(0, len);
        a += CUT_STRING_SYMBOL;
    }
    return a;
}

function menuItem_Select(idx){
    if (idx != null) {
        document.getElementById("td"+idx).style.visibility = "hidden";
        document.getElementById("menu"+idx).style.visibility = "visible";
    }
}

function menuItem_Unselect(idx){
    if (idx != null) {
        document.getElementById("td"+idx).style.visibility = "visible";
        document.getElementById("menu"+idx).style.visibility = "hidden";
    }
}

function getkeydown(e) {
    _debug('getkeydown() keyCode:'+e.keyCode+'; which:'+e.which+ ' alt: '+
e.altKey+ ' ctrlKey: '+e.ctrlKey);

    ec = e.keyCode;
    ew = e.which;
    es = e.shiftKey;

    pat = /^(\\S+)_ (\\S+)/;

    // NOTE!!! This code is necessary to distinguish the codes, which are
generating by RC
    // Accept the agreement:
    //          Ctrl = 1, Alt = 0, keyCode = 32 (Space) : ENTER on the keyboard
and OK on the RC
    //          Ctrl = 0, Alt = 1, keyCode = 32 (Space) : SPACE on the keyboard
and MIC on the RC
    if (ec == 32 && e.ctrlKey && !e.altKey) {
        ec = 13;
        ew = 13;
    }
}
```

```
if (CHECK_ALT_CTRL) {
    altCtrl = e.altKey ;//&& e.ctrlKey;
}
else{
    altCtrl = 1;
}

/*if(altCtrl){
    ec = 0;
}
else {
    if(e.ctrlKey){
        ew=0;
    }
    else{
        if(ec > 90 && ew != 0){
            ec = 0;
        }
    }
}*/

if (altCtrl && ew == 117) { // "Power" button
    if (!inStandBy) { // Switching off
        if (timerRedirect){
            clearTimeout(timerRedirect);
            timerRedirect = null;
            repeatTimer = 'Redirect';
            repeatTimeout = true;
        }
        if (timerToMenu){
            clearTimeout(timerToMenu);
            timerToMenu = null;
            repeatTimer = 'ToMenu';
            repeatTimeout = true;
        }
    }
    else{ // Switching on
        if (repeatTimeout){
            switch (repeatTimer){
                case 'Redirect':
                    timerRedirect = setTimeout(redirect, SERVICE_PRESS_INTERVAL);
                    break;
                case 'ToMenu':
                    timerToMenu = setTimeout(init_continue, SERVICE_PRESS_INTERVAL);
                    break;
            }
            repeatTimeout = false;
            repeatTimer = null;
        }
    }
}
```

```
inStandBy = !inStandBy;
if (!STB_EMULATION){
    stb_OnOff(inStandBy);
    stb.StandBy(inStandBy);
}
return;
}

if (inStandBy) {
    CONTINUE = false;
    return;
}

switch (ec){
    case 38: // Up
    {
        if (curMenuIdx) {
            menuItem_Unselect(curMenuIdx);
            curMenuIdx--;
            menuItem_Select(curMenuIdx);
        }
        break;
    }
    case 40: // Down
    {
        if (curMenuIdx < rowsTotal-1) {
            menuItem_Unselect(curMenuIdx);
            curMenuIdx++;
            menuItem_Select(curMenuIdx);
        }
        break;
    }
    case 13: // OK
        gotoPage(curMenuIdx);
        break;

    case 120: // "Services"
        if(!servicePressed){
            infoButtonPressed();
        }
        break;
}

if (CFG_PARAM_DEBUG) {
    switch (ew) {
        case 113: // Debug "Info" by "Q". This code doesn't work with RC
            infoButtonPressed();
            break;
    }
}
}
```

```
}

function infoButtonPressed(){
  if (timerRedirect){
    clearTimeout(timerRedirect);
    timerRedirect = null;
  }
  if (timerToMenu){
    clearTimeout(timerToMenu);
    timerToMenu = null;
  }
  //document.getElementById("menu_container").innerHTML = SERVICE_LOADING;
  document.getElementById("gotoServ").innerHTML = "";
  beforeLoadingPortal(SERVICE_LOADING);
  sm_Message = '';
  servicePressed = true;
  _debug('servicePressed = '+servicePressed);
  setTimeout(init_continue,100);
}
// menuIdx = index from 0 in menu
function gotoPage(menuIdx){
  beforeLoadingPortal(PORTAL_LOADING);
  var url = eval("portal_"+(menuIdx+1));
  var p = getProtoAndHostname(url);
  if (!p.protocol) {
    url = 'http://' + url;
  }
  //document.location = 'http://'+serv_ip+'/'+portal+'/index.html';
  _debug("HERE !!!! > "+url);
  location.href = url;
}

</script>
</head>

<body onload="loader()" onKeyPress="getkeydown(event)">
<div id="pageIndex">
<table align="center" width="630" height="420" style="table-layout:fixed;">
<tr align="center" valign="middle">
  <td height="400" id="menu_container"></td>
</tr>
<tr>
  <td height="*"></td>
</tr>
<tr>
  <td id="gotoServ" class="Verdana14Grey" height="50" align="center"></td>
</tr>
</table>
</div>
<!-- Modal window -->
```

```
<div id="pad" align="center"></div>
<div id="msgWindow" align="center"></div>
<!-- /Modal window -->
<div id="xpcom" style="margin:2px;"></div>
<div id="emul"></div>

</body>
</html>
```

</spoiler>

From:
<https://docs.infomir.com.ua/> -

Permanent link:
https://docs.infomir.com.ua/doku.php?id=en:stb_webkit:faq:disable_portal_selection_page

Last update: **2019/05/17 11:23**

