

//

```
//Version: 3.1
//=====
// en: ADD SOME EXTRA BUTTONS TO THE EDITPANEL
// de: FÜGE NEUE BUTTON IN DIE WERKZEUGLEISTE
//=====
// Vorschläge für neue Buttons werden gerne entgegengenommen
// Die Reihenfolge und Anzahl der Buttons ist über die (alphabetische) Variable XEBOrder wählbar.

//=====
//Control Variables
//
//rmEditButtons - Removes standard toolbar buttons
//XEBOrder - The order in which the buttons are displayed

document.write('<link rel="stylesheet" type="text/css" href="' +
                'http://en.wikipedia.org/w/index.php?title=User:MarkS/XEB/live.css' +
                '&action=raw&ctype=text/css&dontcountme=s">');

if(typeof XEBPopups== 'undefined')XEBPopups=true;
if(typeof XEBHideDelay== 'undefined')XEBHideDelay=0.5; //Time before the popup disappears after the mouse moves out
if(typeof XEBExtendEditSummary == 'undefined')XEBExtendEditSummary=true; // Is the edit summary extended after a popup

//fills the variable mwCustomEditButtons (s. function in /wikibits.js), with buttons for the Toolbar
function addCustomButton(imageFile, speedTip, tagOpen, tagClose, sampleText){
mwCustomEditButtons.push({
    "imageFile": imageFile,
    "speedTip": speedTip,
    "tagOpen": tagOpen,
    "tagClose": tagClose,
    "sampleText": sampleText});
}

if (typeof usersignature == 'undefined') var usersignature = '-- ~\~\~\~';

var Isrc='http://upload.wikimedia.org/wikipedia/commons/';

// English Wikipedia creates 11 extra buttons which are stored in mwCustomEditButtons
// rather than mwEditButtons. However, there is no guarantee it will always be 11
// so we count them here.
var enExtraButtons=mwCustomEditButtons.length;

var BDict={
'A':['e/e9/Button_headline2.png','Secondary headline','\n===','===','Secondary headline'],
'B':['1/13/Button_enter.png','Line break','<br />','',''],
'C':['5/5f/Button_center.png','Center','<div style="text-align: center;">\n','\n<\div>','Centred text'],
'D':['e/ea/Button_align_left.png','Left-Align','<div style="text-align: left; direction: ltr; margin-left: 1em;">\n','\n<\div>','Left-align text'],
'D1':['a/a5/Button_align_right.png','Right-Align','<div style="text-align: right; direction: ltr; margin-left: 1em;">\n','\n<\div>','Right-align text'],
'E':['0/04/Button_array.png','Table','\n| class="wikitable" \n|- \n| 1 || 2\n|- \n| 3 || 4','\n|}\n',''],
'F':['1/1e/Button_font_color.png','Insert coloured text','<span style="color: ','">Coloured text<\span>','ColourName'],
'FS':['1/1b/Button_miss_signature.png','Unsigned post','{{subst:unsigned|','|date}}','user name or IP'],
'G':['9/9e/Btn_toolbar_gallery.png','Picture gallery','\n<gallery>\nImage:""[[M63]]\nImage:Mona Lisa.jpg|[[Mona Lisa]]\nImage:True',''],
'H':['7/74/Button_comment.png','Comment','","','Comment'],
'I1':['6/6a/Button_sup_letter.png','Superscript','<sup>','<\sup>','Superscript text'],
'I2':['a/aa/Button_sub_letter.png','Subscript','<sub>','<\sub>','Subscript text'],
'J1':['5/58/Button_small.png','Small','<small>','<\small>','Small Text'],
'J2':['5/56/Button_big.png','Big text','<big>','<\big>','Big text'],
'K':['b/b4/Button_category03.png','Category','[[Category:""]],'Category name'],
'L':['8/8e/Button_shifting.png','Insert tab(s)','','',''],
'M':['f/fd/Button_blockquote.png','Insert block of quoted text','<blockquote style="border: 1px solid blue; padding: 2em;">\n','\n<\blockquote>'],
'N':['4/4b/Button_nbsp.png','nonbreaking space',' ',' ',''],
'O':['2/23/Button_code.png','Insert code','<code>','<\code>','Code'],
'P':['3/3c/Button_pre.png','Pre formatted Text','<pre>','<\pre>','Pre formatted text'],
'P1':['9/93/Button_sub_link.png','Insert link to sub-page','[[','/Sub_Page]]','Page'],
'Q':['d/d3/Button_definition_list.png','Insert definition list','\n; ','\n: Item 1\n: Item 2','Definition'],
'R':['7/79/Button_reflink.png','Insert a reference','<ref>','<\ref>','Insert reference material'],
'R1':['7/79/Button_reflink.png','Start a reference','<ref name="','','Reference name'],
'R2':['9/99/Button_reflink_advanced2.png','Insert reference material','">','</ref>','Reference material'],
}
```

```

'R3': ['1/1a/Button_reflink_advanced_3.png', 'No reference material', '', '</>', ''],
'R4': ['9/9a/Button_references.png', 'Reference footer', "\n==Notes==\n\n<div class='references-small'\>\n<references/>\n</div>", '', ''],
'S': ['c/c9/Button_strike.png', 'Strikeout', '<s>', '</s>', 'Struck out text'],
'T': ['e/eb/Button_plantilla.png', 'Template', '{{', '}}', 'Template name'],
'TS': ['a/a4/TableStart.png', 'Start a table', '{', '}', ''],
'TC': ['7/71/TableCell.png', 'Table cell', '|', '}', ''],
'TE': ['0/06/TableEnd.png', 'End a table', '}', '}', ''],
'TR': ['4/4c/TableRow.png', 'Start a table row', '|-', '}', ''],
'T1': ['3/30/Tt_icon.png', 'Teletype text', '<tt>', '</tt>', 'Teletype Text'],
'TL': ['3/37/Button_tl_template.png', 'Template link', "{{subst:"+ "tl|", "}}", 'Template name'],
'U': ['f/fd/Button_underline.png', 'Underlined', "<u>", "</u>", 'Underlined text'],
'V': ['c/c8/Button_redirect.png', 'Redirect', "#REDIRECT [[", "]]", 'Article Name'],
'W': ['8/88/Btn_toolbar_enum.png', 'Numbering', "\n# ", "\n# Element 2\n# Element 3", 'Element 1'],
'X': ['1/11/Btn_toolbar_liste.png', 'List', "\n* ", "\n* Element B\n* Element C", 'Element A'],
'Y1': ['c/ce/Button_no_include.png', 'No Include', "<noinclude>", "</noinclude>", 'Text'],
'Y2': ['7/79/Button_include.png', 'Include only', "<includeonly>", "</includeonly>", 'Text'],
'Z': ['3/35/Button_substitute.png', 'Substitute', "{{subst:", "}}", 'Template'],
'AI': ['1/1c/Button_advanced_image.png', 'Advanaced Image', "[[Image:", "|thumb|right|px|Caption]]", 'FileName.jpg'],
'GEO': ['b/b8/Button_Globe.png', 'Geo location', "", "", ""],
'TALK': ['4/49/Button_talk.png', 'Add talk template', "", "", ""]
];

var XEBOrder2=[];

addOnloadHook(initButtons);
if(!wgIsArticle)// only if edit
{

    if(XEBPopups)hookEvent("load", extendButtons);
}

function initButtons(){

    var bc,d;

    if (typeof XEBOrder!='string') // can be modified
        XEBOrder2="A,D,C,Dl,F,U,Jl,E,G,Q,W,X,K,L,H,O,R,T".split(",");
    else if (XEBOrder.toLowerCase()=='all')
        for (b in BDict) XEBOrder2.push(b);
    else XEBOrder2=XEBOrder.toUpperCase().split(",");

    for (b in BDict) BDict[b][0] = Isrc+BDict[b][0]; // // Add the start of the URL (Isrc) to the XEB buttons
    // If the user has defined any buttons then add them into the available button lists

    if (typeof myButtons=='object')
        for (b in myButtons) BDict[b] = myButtons[b]; // custom user buttons
    // Add the media wiki standard buttons into the available buttons

    for (b in mwEditButtons) { // add standard buttons for full XEB order changing

        //      BDict[b]=[];
BDict[b]=[mwEditButtons[b].imageFile,mwEditButtons[b].speedTip,mwEditButtons[b].tagOpen,mwEditButtons[b].tagClose,mwEditButtons[b].s

//      for (d in mwEditButtons[b]) BDict[b].push(mwEditButtons[b][d]);
    }

    // Build the new buttons

    for (i=0;i<XEBOrder2.length;i++) {
        bc = BDict[XEBOrder2[i]];

        //Check if bc is an object
        // - protects if user specified a non-existant buttons
        // - IE causes a javascript error when viewing a page
        if(typeof bc=='object')
        {

            //Call addCustomButton in wikibits
            addCustomButton(bc[0],bc[1],bc[2],bc[3],bc[4]);

        }
    }
}

```

```

    }

    // Remove the default buttons (if requested by the user)
    eraseButtons();
}

/** en: Removes arbitrary standard buttons from the toolbar
 * @author: [[de:User:Olliminatore]]
 * @version: 0.1 (01.10.2006) */

function eraseButtons(){

    //Remove the buttons the user doesn't want

    if(typeof rmEditButtons!='object') return;

    if (typeof rmEditButtons[0] == 'string' && rmEditButtons[0].toLowerCase() == 'all')
    {
        mwEditButtons=[];
        for(i=0;i<nExtraButtons;i++){mwCustomEditButtons.shift();}
    }
    //Sort the user's requests so we remove the button with the highest index first
    //- This ensures we remove the buttons the user expects whatever order he requested the buttons in
    rmEditButtons.sort(sortit);

    //Remove individual buttons the user doesn't want

    for(i=0;i<rmEditButtons.length;i++){
        var n=rmEditButtons[i];
        //Standard Wikimedia buttons
        if(n>=0 && n<mwEditButtons.length){
            if(n<mwEditButtons.length){
                var x = -1;
                while(++x<mwEditButtons.length)
                    if(x>=n)
                        mwEditButtons[x] = mwEditButtons[x+1];
            }
            mwEditButtons.pop();
        }
        //Extra buttons in English Wikipedia
        n=n-mwEditButtons.length;
        if(n>0 && n<mwCustomEditButtons.length){
            if(n<mwCustomEditButtons.length){
                var x = -1;
                while(++x<mwCustomEditButtons.length)
                    if(x>=n)
                        mwCustomEditButtons[x] = mwCustomEditButtons[x+1];
            }
            mwCustomEditButtons.pop();
        }
    }
};

//Function:
//    sortit
//Purpose:
//    Used to sort the rmEditButtons array into descending order
function sortit(a,b){
    return(b-a)
}

//Function:
//Purpose:
//    Adds extended onclick-function to some buttons
function extendButtons(){

    if(!(allEditButtons = document.getElementById('toolbar'))) return false;
    if(typeof editform != 'undefined')
        if(!(window.editform = document.forms['editform'])) return false;

    // table

```

```

extendAButton(Isrc+"0/04/Button_array.png",XEBPopupTable)
extendAButton(Isrc+"7/79/Button_reflink.png",XEBPopupRef)
extendAButton(Isrc+"b/b8/Button_Globe.png",XEBPopupGeoLink)
extendAButton(Isrc+"4/49/Button_talk.png",XEBPopupTalk)
extendAButton(Isrc+"1/1c/Button_advanced_image.png",XEBPopupImage)
//extendAButton(Isrc+"6/6a/Button_sup_letter.png",XEBPopupFormattedText)

// redirect -##IE doesn't like this line. Object doesn't support this property or method
//c=XEBOrder2.getIndex('V');

//      if(c != -1)
//      allEditButtons[bu_len+c].onclick=function(){
//      var a='#REDIRECT \[\'+'prompt("Which page do you want to redirect to?\")+\'\\'];
//      document.editform.elements['wpTextbox1'].value=a;
//      document.editform.elements['wpSummary'].value=a;
//      document.editform.elements['wpWatchthis'].checked=false
//  };
};

function extendAButton(url,newfunc)
{
    if(!(allEditButtons = document.getElementById('toolbar'))) return false;
    if(typeof editform != 'undefined')
        if(!(window.editform = document.forms['editform'])) return false;
    allEditButtons = allEditButtons.getElementsByTagName('img');
    for(i=0;i<allEditButtons.length;i++)
    {
        if(allEditButtons[i].src==url)
        {
            allEditButtons[i].onclick=newfunc;
        }
    }
}

//=====
// General purpose popup code
//=====

function getXEBPopupDiv(name)
{
    XEBMainDiv= document.getElementById("XEB");
    if(XEBMainDiv==null){
        XEBMainDiv=document.createElement("div");
        document.body.appendChild(XEBMainDiv);
        XEBMainDiv.id="XEB";
    }

    me= document.getElementById("XEBPopup" & name);
    if(!(me==null)) return me;
    me=document.createElement("div");
    XEBMainDiv.appendChild(me);

    me.id="XEBPopup";
    me.style.position='absolute';
    me.display='none';
    me.visibility='hidden';
    me.onmouseout=CheckHideXEBPopup;
    me.onmouseover=cancelHidePopup;
    return me;
}

//Function:
//      CheckHideXEBPopup
//Purpose:
//      Looks at the cursor position and if it has moved outside the popup it will close the popup
//Called:
//      When the onMouseEvent is fired on the popup

function CheckHideXEBPopup(e){
    m= document.getElementById("XEBmnu");
    if(is_gecko)
    {
        ph=m.offsetHeight;
    }
}

```

```

        var x=e.clientX + window.scrollX;
        var y=e.clientY + window.scrollY;;
        s=window.getComputedStyle(m,"");
        ph=s.height;
        ph=Number(ph.substring(0,ph.length-2));
    }
    else
    {
        var x=event.clientX+ document.documentElement.scrollLeft + document.body.scrollLeft;
        var y=event.clientY+ document.documentElement.scrollTop + document.body.scrollTop;
        ph=m.offsetHeight;

        pl=curPopup.x;
        pt=curPopup.y;
        pw=m.style.width;
        pw=Number(pw.substring(0,pw.length-2));

        if (x>(pl+2) && x<(pl+pw-5) && y>(pt+2) && y<(pt+ph-5)) return;
        curPopup.hideTimeout=setTimeout('hideXEBPopup()', XEBHideDelay*1000);
    }

function cancelHidePopup()
{
    clearTimeout(curPopup.hideTimeout)
}

function hideXEBPopup(){
    XEBMainDiv= document.getElementById("XEB");
    m= document.getElementById("XEBPopup");
    XEBMainDiv.removeChild(m);
}

function XEBstartDrag(e)
{
    m=new GetPos(e||event);
    curPopup.startDrag.mouse=m;
    curPopup.startDrag.floatpopup.y=parseInt(curPopup.div.style.top);
    curPopup.startDrag.floatpopup.x=parseInt(curPopup.div.style.left);
    curPopup.dragging=true;
}

function XEBstopDrag(e)
{
    if (curPopup.dragging==false) return;
    curPopup.dragging=false;
}

function XEBDrag(e)
{
    if (curPopup.dragging==false) return;

    m=new GetPos(e||event);
    x=parseInt(curPopup.startDrag.floatpopup.x+(m.x-curPopup.startDrag.mouse.x));
    y=parseInt(curPopup.startDrag.floatpopup.y+(m.y-curPopup.startDrag.mouse.y));

    curPopup.div.style.top=y+"px";
    curPopup.div.style.left=x+"px";

    curPopup.x=x;
    curPopup.y=y;
}

//=====
// Popup: Table
//=====

function XEBPopup(name,x,y)
{
    // Make sure the popup can appear on the screen

    this.IESelectedRange=XEBgetIESelectedRange();

    winW=(is_gecko)?window.innerWidth:document.body.offsetWidth;

```

```

        if ((winW-this.width)<x)x=(winW-this.width);

        this.div=getXEBPopupDiv(name);
        this.div.style.zIndex=2000;
        this.div.display="inline";
        this.div.visibility="visible";
        this.div.style.top=y + "px";
        this.x=x;
        this.y=y;
        this.name=name;

        this.startDrag=new Object;
        this.startDrag.floatpopup=new Object;
    }

function setInnerHTML(text)
{
    winW=(is_gecko)?window.innerWidth:document.body.offsetWidth;
    if ((winW-this.width)<this.x)this.x=(winW-this.width);
    this.div.style.left=this.x+ "px";

    mt="<div id='XEBmnu' style='width:" + this.width + "px' >";
    mt+="
```

```

        + '<i>The default table allows for fields and values.</i><p\>'
        + 'Check "Item column" to allow for the table to have fields, items, and values.</i><p\>'
        + '<p><button onClick="javascript:insertTableCode()">Insert</button>'
        + '<button onClick="hideXEBPopup()">Cancel</button>'

    curPopup.setInnerHTML(mt);

    return true;
}

function insertTableCode(){
    f=document.XEBPopupTableForm;
    var caption = (f.inputCaption.checked)?'|+ TABLE CAPTION \n':"";
    var exhead = (f.inputHead.checked)?'| - style="background: #DDFFDD;\n':"";
    var nbRow = parseInt(f.inputRow.value);
    var nbCol = parseInt(f.inputCol.value);
    var exfield = f.inputItems.checked;
    var align = (f.inputAlign.checked)?'align="center"':"";

    //generateTable(caption, exhead, nbCol, nbRow, exfield, align);

    var code = "\n";
    code += '{| {prettytable}} ' + align + ' '; // en: class="wikitable"
    code+=(f.inputSort.checked)?'class="sortable" \n':'\n';
    code += caption + exhead;
    if (exfield) code += '! \n';
    for (i=1;i<nbCol+1;i++) code += '! FELD ' + i + ' \n';
    var items = 0;
    for (var j=0;j<nbRow;j++){
        if (exfield) {
            items++;
            code += '| - \n! style="background: #FFDDDD;"|ITEM ' + items + ' \n';
        } else code += '| - \n';
        for (i=0;i<nbCol;i++) code += '| Element \n';
    }
    code += '|\n';
    hideXEBPopup();
    insertTags('', '', code);
    extendSummary('table');

    return false;
}

// Get the text currently selected by user in the textAra
// This code is based on part of the insertTags function in wikibits.js

function XEBGetSelectedText()
{
    var txtarea;
    if (document.editform) {
        txtarea = document.editform.wpTextbox1;
    } else {
        // some alternate form? take the first one we can find
        var areas = document.getElementsByTagName('textarea');

        txtarea = areas[0];
    }
    // IE & Opera
    if (document.selection && !is_gecko)
    {
        var theSelection = document.selection.createRange().text;
        if (!theSelection) theSelection='';
    }
    // Mozilla
    else if (txtarea.selectionStart || txtarea.selectionStart == '0') {
        var replaced = false;
        var startPos = txtarea.selectionStart;
        var endPos = txtarea.selectionEnd;
        var theSelection = (txtarea.value).substring(startPos, endPos);
        if (!theSelection) theSelection='';
    }
    return theSelection;
}

```

```

//Notes:
//      IE loses the cursor position in the textarea when the popup is used.
//      So we save the cursor position here
function XEBgetIESelectedRange(){
    var IESel=new Object;
    var txtarea;
    if (document.editform) {
        txtarea = document.editform.wpTextbox1;
    } else {
        // some alternate form? take the first one we can find
        var areas = document.getElementsByTagName('textarea');

        txtarea = areas[0];
    }
    // IE & Opera

    if (document.selection  && !is_gecko)
    {
        txtarea.focus();
        IESel.Rng=document.selection.createRange();
        return IESel;
    }
}

function XEBinsertText(beforeText,selText,afterText,IESelectedRange) {
    var newText=beforeText + selText + afterText;
    var txtarea;
    if (document.editform) {
        txtarea = document.editform.wpTextbox1;
    } else {
        // some alternate form? take the first one we can find
        var areas = document.getElementsByTagName('textarea');
        txtarea = areas[0];
    }

    // IE
    if (document.selection  && !is_gecko) {

        tr=IESelectedRange.Rng;
        tr.text=newText;
        txtarea.focus();
        //txtarea.caretpos=tr.duplicate();
        tr.select();

        return;

    // Mozilla
    } else if(txtarea.selectionStart || txtarea.selectionStart == '0') {
        var replaced = false;
        var startPos = txtarea.selectionStart;
        var endPos = txtarea.selectionEnd;

        if (endPos-startPos) {
            replaced = true;
        }
        var scrollTop = txtarea.scrollTop;
        var myText = (txtarea.value).substring(startPos, endPos);
        //
        // if (!myText) {
        //     myText=sampleText;
        //
        // }
        // if (myText.charAt(myText.length - 1) == " ") { // exclude ending space char, if any
        //     subst = tagOpen + myText.substring(0, (myText.length - 1)) + tagClose + " ";
        // } else {
        //     subst = tagOpen + myText + tagClose;
        //
        // }
        txtarea.value = txtarea.value.substring(0, startPos) + newText +
            txtarea.value.substring(endPos, txtarea.value.length);
        txtarea.focus();
        //set new selection
        if (!replaced) {
            var cPos = startPos+(newText.length);
            txtarea.selectionStart = cPos;

```



```

        txtarea.selectionEnd = cPos;
    } else {
        txtarea.selectionStart = startPos+beforeText.length;
        txtarea.selectionEnd = startPos+beforeText.length+selText.length;
    }
    txtarea.scrollTop = scrollTop;

    // All other browsers get no toolbar.
    // There was previously support for a crippled "help"
    // bar, but that caused more problems than it solved.
    }
    // reposition cursor if possible
    if (txtarea.createTextRange) {

        txtarea.caretPos = document.selection.createRange().duplicate();
    //txtarea.caretPos = IESelectedRange.Rng;
    }
    txtarea.focus();
}

//=====
// Table generator
//=====
/** en: Generate an array using Mediawiki syntax
 * @author: originally from fr:user:dake
 * @version: 0.2 */
function generateTable(caption, exhead, nbCol, nbRow, exfield, align){

};

function XEBPopupRef(e){

    m=new GetPos(e||event);

    curPopup=new XEBPopup("ref",m.x,m.y);
    curPopup.width=500;
    mt='<p>Enter the reference parameters below: </p>'
        +'<form name="XEBPopupRefForm">'
        +'Name:<input type="text" name="refName" value="" size="10"><p\>'
        +'Material:<input type="text" name="refMaterial" value=" ' + XEBGetSelectedText() + ' " size="20">'
        +'</form>'
        +'<p><button onClick="javascript:insertRef()">Insert</button>'
        +'<button onClick="hideXEBPopup()">Cancel</button>';

    curPopup.setInnerHTML(mt);
    // document.XEBPopupRefForm.refName.focus();
    return true;
}

function insertRef(){
    f=document.XEBPopupRefForm;
    var refName = f.refName.value;
    var refMaterial=f.refMaterial.value;

    hideXEBPopup();
    var code1='<ref';
    code1+=(refName)?' name="'+refName+'>':'>';
    code2=refMaterial;
    code3='</ref>'
    XEBinsertText (code1,code2,code3,curPopup.IESelectedRange);

    extendSummary('ref');
    return false;
}

//===GEO LINK Function=====

function XEBPopupGeoLink(e)
{
    m=new GetPos(e||event);

```

```

curPopup=new XEBPopup("geo",m.x,m.y);
curPopup.width=300;
mt='<p>Enter the location parameters below: </p>'
+'<form name="XEBPopupGeoLinkForm">'
+'Loction:<p\>'
+'<table style="background: transparent;">'
+'<tr><td>Latitude:</td><td><input type="text" autocomplete="off" name="geoLatDeg" value="" size="4"></td>'
+'<td><input type="text" name="geoLatMin" size="4"></td>'
+'<td><input type="text" name="geoLatSec" size="4"></td>'
+'<td><select name="geoLatNS"><option value="N">N<option value="S">S</select></td></tr>'
+'<tr><td>Longitude:</td><td><input type="text" name="geoLonDeg" value="" size="4"></td>'
+'<td><input type="text" name="geoLonMin" value="" size="4"></td>'
+'<td><input type="text" name="geoLonSec" value="" size="4"></td>'
+'<td><select name="geoLonEW"><option value="E">E<option value="W">W</select></td></tr>'
+'</table>'
+'Region:<input type="text" name="geoRegion" value="" size="4"><p\>'
+'Type:'
+'<SELECT NAME="geoType" size="5">'
+'<OPTION VALUE="country">Country<OPTION VALUE="state">State'
+'<OPTION VALUE="admlst">Admin unit, 1st level<OPTION VALUE="adm2st">Admin unit, 2nd level'
+'<OPTION VALUE="city">City<OPTION VALUE="airport">Airport'
+'<OPTION VALUE="mountain">Mountain<OPTION VALUE="isle">Isle'
+'<OPTION VALUE="waterbody">Waterbody<OPTION VALUE="landmark" SELECTED>Landmark'
+'<OPTION VALUE="forest">forest</SELECT><br>'
+'Title: <input type="checkbox" name="geoTitle" ><p\>'
+'</form>'
+'<p><button onClick="javascript:insertGeoLink()">Insert</button>'
+'<button onClick="hideXEBPopup()">Cancel</button>';

curPopup.setInnerHTML(mt);
document.paramForm.refName.focus();
return true;

}

function insertGeoLink()
{
    f=document.XEBPopupGeoLinkForm;

    var code='{{Coor ';
    if(f.geoTitle.checked)code+='title ';
    ft='dms';
    if(f.geoLatSec.value=='&&f.geoLonSec.value=='')ft='dm';
    if(ft=='dm'&&f.geoLatMin.value=='&&f.geoLonMin.value=='')ft='d';
    code+=ft;
    code+='|'+f.geoLatDeg.value;
    code+=(ft=='dm'||ft=='dms')?'|'+f.geoLatMin.value:'';
    code+=(ft=='dms')?'|'+f.geoLatSec.value:'';
    code+='|'+f.geoLatNS.value;
    code+='|'+f.geoLonDeg.value;
    code+=(ft=='dm'||ft=='dms')?'|'+f.geoLonMin.value:'';
    code+=(ft=='dms')?'|'+f.geoLonSec.value:'';
    code+='|'+f.geoLonEW.value;
    code+='|type:'+f.geoType.value+'_region:'+f.geoRegion.value;
    code+='}}';
    insertTags('','', code);
    extendSummary('geo-location');
    hideXEBPopup();
    return false;
}

//====Talk Page entry Function=====

function XEBPopupTalk(e)
{
    m=new GetPos(e||event);

    curPopup=new XEBPopup("talk",m.x,m.y);
    curPopup.width=200;
    mt='<div style="font-size:medium"><p>Please choose:</p>'
    mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(1)">Test1</span><br>'
    mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(2)">Self Test</span><br>'
    mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(3)">Nonsense</span><br>'
    mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(4)">Please stop</span><br>'

```

```

mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(5)">Last chance</span><br>'
mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(6)">Blanking</span><br>'
mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(7)">Blatant</span><br>'
mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(8)">*BLOCKED*</span><br>'
mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(9)">Spam</span><br>'
mt+=''<span class="XEBMnuItm" onclick="XEBInsertTalk(10)">Npov</span></div>'

curPopup.setInnerHTML(mt);

return true;

}

function XEBInsertTalk(itm)
{
    hideXEBPopup();
    if(itm==1)code='{{subst:test1-n|}}';
    if(itm==2)code='{{subst:selftest-n|}}';
    if(itm==3)code='{{subst:test2-n|}}';
    if(itm==4)code='{{subst:test3-n|}}';
    if(itm==5)code='{{subst:test4-n|}}';
    if(itm==6)code='{{subst:test2a-n|}}';
    if(itm==7)code='{{subst:bv-n|}}';
    if(itm==8)code='{{subst:blatant|}}';
    if(itm==9)code='{{subst:spam-n|}}';
    if(itm==10)code='{{subst:NPOV user}}';

    insertTags('',' ', code);
    return false;
}

function XEBPopupImage(e)
{
    m=new GetPos(e||event);

    curPopup=new XEBPopup("image",m.x,m.y);
    curPopup.width=300;

    mt='<p>Enter the image parameters below: </p>'
        +'<form name="XEBPopupImageForm">'
        +'File:<input type="text" name="imgFile" value="' + XEBGetSelectedText() + '" size="30"><br>'
        +'Type:<SELECT NAME="imgType">'
        +'<OPTION VALUE="thumb">Thumbnail'
        +'<OPTION VALUE="frame">Frame'
        +'<OPTION VALUE="none">[not specified]'
        +'</SELECT><br>'
        +'Location:<SELECT NAME="imgLocation">'
        +'<OPTION VALUE="left">Left'
        +'<OPTION VALUE="center">Centre'
        +'<OPTION VALUE="right">Right'
        +'<OPTION VALUE="none">None'
        +'</SELECT><br>'
        +'Size:<input type="text" name="imgSize" value="100" size="3">px<br>'
        +'Caption:<input type="text" name="imgCaption" value="" size="30"></p>'
        +'</form>'
        +'<p><button onClick="javascript:XEBInsertImage()">Insert</button>'
        +'<button onClick="hideXEBPopup()">Cancel</button>';

    curPopup.setInnerHTML(mt);

    return true;
}

function XEBInsertImage()
{
    f=document.XEBPopupImageForm;
    hideXEBPopup();
    var code='[[Image:';
    code+=f.imgFile.value;
    code+='|'+f.imgType.value;
    code+='|'+f.imgLocation.value;
    code+='|'+f.imgSize.value;
    code+='|'+f.imgCaption.value;
    code+=']]';
    insertTags('',' ', code);
    extendSummary('image');
}

```

```

        return false;
    }

function XEBPopupFormattedText(e)
{
    m=new GetPos(e||event);

    curPopup=new XEBPopup("image",m.x,m.y);
    curPopup.width=300;

    mt='<form name="XEBPopupImageForm">'
        +'<table style="background: transparent;">'
        +'<tr><td>Bold:</td><td><input type="checkbox" name="textBold"></td>'
        +'<td>Superscript:</td><td><input type="checkbox" name="textSuperscript"></td></tr>'
        +'<tr><td>Italic:</td><td><input type="checkbox" name="textItalic"></td>'
        +'<td>Subscript:</td><td><input type="checkbox" name="textSubscript"></td></tr>'
        +'<tr><td>Strike:</td><td><input type="checkbox" name="textStrike"></td>'
        +'<td> </td></tr>'
        +'</table>'
        +'Size:<SELECT NAME="textSize">'
        +'<OPTION VALUE="small">small'
        +'<OPTION VALUE="normal">[Normal]'
        +'<OPTION VALUE="big">big'
        +'</SELECT><br><table style="background:transparent;"><tr><td>Colour:</td><td>'
        +'<table width="100px">'
        +'<tr><td colspan="4">None</td></tr>'
        +'<tr><td bgcolor="aqua"> </td><td bgcolor="gray"> </td>'
        +'<td bgcolor="olive"> </td><td bgcolor="navy"> </td></tr>'
        +'<tr><td bgcolor="black"> </td><td bgcolor="green"> </td>'
        +'<td bgcolor="purple"> </td><td bgcolor="teal"> </td></tr>'
        +'<tr><td bgcolor="blue"> </td><td bgcolor="lime"> </td>'
        +'<td bgcolor="red"> </td><td bgcolor="white"> </td></tr>'
        +'<tr><td bgcolor="fuchsia"> </td><td bgcolor="maroon"> </td>'
        +'<td bgcolor="silver"> </td><td bgcolor="yellow"> </td></tr>'
        +'</table></td></tr>'
        +'</form>'
        +'Sample:'
        +'<span id="sampleText">Text</span>'
        +'<p><button onClick="javascript:XEBInsertFormattedText()">Insert</button>'
        +'<button onClick="hideXEBPopup()">Cancel</button>';

    curPopup.setInnerHTML(mt);

    return true;
}

function XEBUpdateSampleText()
{
    f=document.XEBPopupImageForm;
}

//=====

function XEBMenuMouseOut(e)
{
    var targ;
    if (!e) var e = window.event;
    if (e.target) targ = e.target;
    else if (e.srcElement) targ = e.srcElement;

    targ.style.color='black';
}

function XEBMenuMouseOver(e)
{
    var targ;
    if (!e) var e = window.event;
    if (e.target) targ = e.target;
    else if (e.srcElement) targ = e.srcElement;

    targ.style.color='red';
}

```

```

//=====
// Other functions
//=====

function XEBgetElementsByClassName(parent,clsName,htmltag){
    var arr = new Array();
    var elems = parent.getElementsByTagName(htmltag);
    for ( var cls, i = 0; ( elem = elems[i] ); i++ ){
        if ( elem.className == clsName ){
            arr[arr.length] = elem;
        }
    }
    return arr;
}

function extendSummary(newText)
{
    if(!XEBExtendEditSummary)return;
    s=document.editform.elements['wpSummary'].value;
    s+=(s=='')?newText:' '+newText;
    document.editform.elements['wpSummary'].value=s;
}

function bug(msg)
{
    if(wgUserName=='MarkS')alert (msg);
}

//

```