AuthnRequestGenerationWebpage

This is the raw code for the webpage. It can also be found attached as an .html on this article. All credit is due to Dan Malone of Cal Poly San Luis Obispo.

To install, just save this file locally and open it in a web browser: SAMLRequestGenerator.html

SAMLRequestGenerator.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>SAML AuthnRequest Generator</title>
<style type="text/css">
.button {
    border: 1px outset black;
    border-radius:10px;
    background-color: #d1d1e0;
    height:50px;
    width:300px;
    cursor:pointer;
    font-size:larger;
}

.button:hover {
    background-color: #427730;
    color:white;
}

td {
    vertical-align:top;
}
</style>

</head>
<body>
<h1>SAML AuthnRequest Generator</h1>

<h1>Step 1 - Parse SAML Parameters</h1>
This step decodes redirect-based SAML AuthnRequests.  You can skip it if you already have the XML.
<form id=samlSource>

Original Request URL:<br>
<textarea autofocus cols=80 rows=15 id=InputRequestURL placeholder="Paste the source request url here">
</textarea>
<br>
<button class="button" type="button" onclick="inputAll()">Parse the SAML AuthnRequest</button>


</form>
<script>
function inputSAMLRequest() {
    samlRequestURL = document.getElementById('InputRequestURL');
    samlRequestString=document.getElementById('ModifySAMLRequest');

    regex=/[?|&]SAMLRequest=([^&]*)/;
    match=regex.exec(samlRequestURL.value);
    if(match != null) {
        osr=match[1];
        osrDecoded=atob(decodeURIComponent(osr));
        osrInflated=zip_inflate(osrDecoded);
        samlRequestString.value=osrInflated;
//         samlRequestString.value=XMLTree(osrInflated);
    } else {
        samlRequestString.value="";
        samlRequestString.placeholder="SAMLRequest parameter not found";
    }
}
```

```
function inputRelayState() {
    samlRequestURL = document.getElementById('InputRequestURL');
    modifyRelayState=document.getElementById('ModifyRelayState');

    regex=/[?|&]RelayState=([^&]*)/;
    match=regex.exec(samlRequestURL.value);
    if(match != null) {
        modifyRelayState.value=decodeURIComponent(match[1]);
    } else {
        modifyRelayState.value="";
        modifyRelayState.placeholder="RelayState not found";
    }
}

function inputAll() {
    inputSAMLRequest();
    inputRelayState();
}

</script>
</form>

<h1>Step 2 - Modify SAML Parameters</h1>
<form id=samlSetup>

<table>
<tr>
<th>Parameter</th>
<th>Value</th>
<th></th>
</tr>

<tr>
<td>Local Time</td>
<td><input id=localTime type="datetime-local" size=40 readonly placeholder="Will be updated automatically"></td>
<td><label><input type=checkbox id=localTimeCheckbox onclick="clickAutomaticCheckbox(this)" checked>Update from
current time each build</label></td>
</tr>


<tr>
<td>ISODate</td>
<td><input id=isoDate size=40 readonly placeholder="Will be updated automatically"></td>
<td><label><input type=checkbox id=isoDateCheckbox onclick="clickAutomaticCheckbox(this)" checked>Convert from
Local Time each build</label></td>
</tr>

<tr>
<td>ID</td>
<td><input id=uniqueID size=40 readonly placeholder="Will be updated automatically"></td>
<td><label><input type=checkbox id=uniqueIDCheckbox onclick="clickAutomaticCheckbox(this)" checked>Generate
random ID each build</label></td>
</tr>
</table>

<br>
AuthnRequest:
<button type="button" onclick="formatSAMLRequest();">Format XML</button>
<label><input type=checkbox id=WrapModifySAMLRequest onclick="wrapModifySAMLRequest(this)">Wrap</label>
<br>
<textarea wrap="off" cols=80 rows=15 id=ModifySAMLRequest placeholder="Paste the AuthnRequest xml block here">
</textarea>

<br>
<br>
<label><input type=checkbox id=RelayStateCheckbox onclick="relayStateCheckbox(this)" checked>Include RelayState<
/label>

<div id=ModifyRelayStateDiv>
RelayState:<br>
```

```
<textarea cols=80 rows=15 id=ModifyRelayState placeholder="Paste the RelayState parameter here">
</textarea>
</div>

<br>
<br>

<table>
<tr>
<th>Form Parameters</th>
<th></th>
</tr>

<tr>
<td>Endpoint Request Method</td>
<td>
<input type = "radio"
     name = "endpointMethod"
     id = "get"
     value = "GET"
     checked = "checked" />
<label for = "get">GET</label>
<input type = "radio"
     name = "endpointMethod"
     id = "post"
     value = "POST" />
<label for = "post">POST</label>
</td>
</tr>

<tr>
<td>Endpoint URL</td>
<td><input id=endpointURL size=80 readonly placeholder="Will be updated automatically"><br>
<label><input type=checkbox id=endpointURLCheckbox onclick="clickAutomaticCheckbox(this)" checked>Update
Endpoint URL from AuthnRequest Destination each build</label></td>
</tr>

</table>

<br>
<br>

<button class="button" type="button" onclick="buildSAMLRequest()">Build the SAML AuthnRequest</button>
</form>


<br>
<h1>Step 3 - Submit to SAML Endpoint</h1>

<div style="outline: 2px dashed black; padding: 5px; display: inline-block;">
<form id=samlForm method=GET action="/" target="_blank">
SAMLRequest (deflate/base64):<br>
<textarea id=SAMLRequest name=SAMLRequest cols=60 rows=5 onfocus="this.select()">
</textarea>

<br>
<div id=RelayStateDiv>
RelayState:<br>
<textarea id=RelayState name=RelayState cols=60 rows=5 onfocus="this.select()">
</textarea>

<br>
</div>

<input class="button" id=samlFormSubmit type=submit value=Submit>
</form>
</div>

<script>
var placeholder=[];
placeholder["localTime"] = "yyyy-MM-ddThh:mm";
```

```
placeholder["isoDate"] = "yyyy-MM-ddThh:mm:ss.SSSSZ";
placeholder["uniqueID"] = "_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
placeholder["endpointURL"] = "https://idp.example.com/idp/profile/SAML2/Redirect/SSO";

function clickAutomaticCheckbox(checkbox) {
    fieldName = checkbox.id.replace("Checkbox","");
    field = document.getElementById(fieldName);
    if (checkbox.checked) {
        field.readOnly = true;
        field.placeholder = "Will be updated automatically";
    } else {
        field.readOnly = false;
        field.placeholder = placeholder[fieldName];
    }
}

function formatSAMLRequest () {
    modifySAMLRequest=document.getElementById('ModifySAMLRequest');
    modifySAMLRequest.value=beautify.xml(modifySAMLRequest.value);
}

function wrapModifySAMLRequest (checkbox) {
    modifySAMLRequest=document.getElementById('ModifySAMLRequest');
    if(checkbox.checked) {
        modifySAMLRequest.wrap="on";
    } else {
        modifySAMLRequest.wrap="off";
    }
}

function relayStateCheckbox(checkbox) {
    modifyRelayState = document.getElementById("ModifyRelayState");
    modifyRelayStateDiv = document.getElementById("ModifyRelayStateDiv");
    relayState = document.getElementById("RelayState");
    relayStateDiv = document.getElementById("RelayStateDiv");
    if (checkbox.checked) {
        modifyRelayState.disabled = false;
        relayState.disabled = false;
        modifyRelayStateDiv.hidden = false;
        relayStateDiv.hidden = false;
    } else {
        modifyRelayState.disabled = true;
        relayState.disabled = true;
        modifyRelayStateDiv.hidden = true;
        relayStateDiv.hidden = true;
    }
}

function buildSAMLRequest() {
    localTimeInput = document.getElementById('localTime');
    if(document.getElementById('localTimeCheckbox').checked){
        currentTime=new Date(Date.now());
        localTime=currentTime.getFullYear().toString()+'-'+
                  (currentTime.getMonth()+1).toString().padStart(2,'0')+'-'+
                  currentTime.getDate().toString().padStart(2,'0')+'T'+
                  currentTime.getHours().toString().padStart(2,'0')+':'+
                  currentTime.getMinutes().toString().padStart(2,'0')+':'+
                  currentTime.getSeconds().toString().padStart(2,'0');
        localTimeInput.value = localTime;
    }

    isoDateInput=document.getElementById('isoDate');
    if(document.getElementById('isoDateCheckbox').checked){
        n2=new Date(localTimeInput.value);
        isoDate=n2.toISOString();
        isoDateInput.value=isoDate;
    }

    uniqueIDInput=document.getElementById('uniqueID');
    if(document.getElementById('uniqueIDCheckbox').checked){
        id="_"+randomGUID();
```

```
        uniqueIDInput.value=id;
    }

    modifySAMLRequest=document.getElementById('ModifySAMLRequest');
    modifySAMLRequest.value=modifySAMLRequest.value.replace(/(IssueInstant=\")([^\"]*)(\")/,'$1'+isoDateInput.
value+'$3');
    modifySAMLRequest.value=modifySAMLRequest.value.replace(/(ID=\")([^\"]*)(\")/,'$1'+uniqueIDInput.
value+'$3');

    samlRequest=document.getElementById('SAMLRequest');
    samlRequest.value = btoa(zip_deflate(modifySAMLRequest.value));

    document.getElementById('RelayState').value=document.getElementById('ModifyRelayState').value;

    endpointURLInput=document.getElementById('endpointURL');
    if(document.getElementById('endpointURLCheckbox').checked){
        regex=/Destination=\"?([^"]*)\"?/;
        match=regex.exec(modifySAMLRequest.value);
        if(match != null) {
            endpointURLInput.value=match[1];
        } else {
            endpointURLInput.value="";
            endpointURLInput.placeholder="Destination not found";
        }
    }

    document.getElementById("samlForm").action = endpointURLInput.value;

    methodButtons = document.getElementById('samlSetup').elements['endpointMethod']
    for (var i=0, len=methodButtons.length; i<len; i++) {
        if ( methodButtons[i].checked ) { // radio checked?
            document.getElementById("samlForm").method = methodButtons[i].value;
            break;
        }
    }

    document.getElementById("samlFormSubmit").focus();

}
</script>

<!--
Including these inline here to make this all in a single file, but could be
done in an external reference too.
-->

<!--
from http://www.onicos.com/staff/iz/amuse/javascript/expert/
mailto:iz@onicos.co.jp
-->
<script>
var zip_WSIZE=32768,zip_STORED_BLOCK=0,zip_STATIC_TREES=1,zip_DYN_TREES=2,zip_DEFAULT_LEVEL=6,zip_FULL_SEARCH=!
0,zip_INBUFSIZ=32768,zip_INBUF_EXTRA=64,zip_OUTBUFSIZ=8192,zip_window_size=2*zip_WSIZE,zip_MIN_MATCH=3,
zip_MAX_MATCH=258,zip_BITS=16,zip_LIT_BUFSIZE=8192,zip_HASH_BITS=13;zip_LIT_BUFSIZE>zip_INBUFSIZ&&alert("error:
zip_INBUFSIZ is too small"),zip_WSIZE<<1>1<<zip_BITS&&alert("error: zip_WSIZE is too large"),
zip_HASH_BITS>zip_BITS-1&&alert("error: zip_HASH_BITS is too large"),(zip_HASH_BITS<8||258!=zip_MAX_MATCH)
&&alert("error: Code too clever");var zip_free_queue,zip_qhead,zip_qtail,zip_initflag,zip_outcnt,zip_outoff,
zip_complete,zip_window,zip_d_buf,zip_l_buf,zip_prev,zip_bi_buf,zip_bi_valid,zip_block_start,zip_ins_h,
zip_hash_head,zip_prev_match,zip_match_available,zip_match_length,zip_prev_length,zip_strstart,zip_match_start,
zip_eofile,zip_lookahead,zip_max_chain_length,zip_max_lazy_match,zip_compr_level,zip_good_match,zip_nice_match,
zip_dyn_ltree,zip_dyn_dtree,zip_static_ltree,zip_static_dtree,zip_bl_tree,zip_l_desc,zip_d_desc,zip_bl_desc,
zip_bl_count,zip_heap,zip_heap_len,zip_heap_max,zip_depth,zip_length_code,zip_dist_code,zip_base_length,
zip_base_dist,zip_flag_buf,zip_last_lit,zip_last_dist,zip_last_flags,zip_flags,zip_flag_bit,zip_opt_len,
zip_static_len,zip_deflate_data,zip_deflate_pos,zip_DIST_BUFSIZE=zip_LIT_BUFSIZE,zip_HASH_SIZE=1<<zip_HASH_BITS,
zip_HASH_MASK=zip_HASH_SIZE-1,zip_WMASK=zip_WSIZE-1,zip_NIL=0,zip_TOO_FAR=4096,
zip_MIN_LOOKAHEAD=zip_MAX_MATCH+zip_MIN_MATCH+1,zip_MAX_DIST=zip_WSIZE-zip_MIN_LOOKAHEAD,zip_SMALLEST=1,
zip_MAX_BITS=15,zip_MAX_BL_BITS=7,zip_LENGTH_CODES=29,zip_LITERALS=256,zip_END_BLOCK=256,
zip_L_CODES=zip_LITERALS+1+zip_LENGTH_CODES,zip_D_CODES=30,zip_BL_CODES=19,zip_REP_3_6=16,zip_REPZ_3_10=17,
zip_REPZ_11_138=18,zip_HEAP_SIZE=2*zip_L_CODES+1,zip_H_SHIFT=parseInt((zip_HASH_BITS+zip_MIN_MATCH-1)
/zip_MIN_MATCH),zip_outbuf=null,zip_extra_lbits=new Array
(0,0,0,0,0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,0),zip_extra_dbits=new Array
```

```
(0,0,0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,13,13),zip_extra_blbits=new Array
(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,3,7),zip_bl_order=new Array(16,17,18,0,8,7,9,6,10,5,11,4,12,3,13,2,14,1,15),
zip_configuration_table=new Array(new zip_DeflateConfiguration(0,0,0,0),new zip_DeflateConfiguration(4,4,8,4),
new zip_DeflateConfiguration(4,5,16,8),new zip_DeflateConfiguration(4,6,32,32),new zip_DeflateConfiguration
(4,4,16,16),new zip_DeflateConfiguration(8,16,32,32),new zip_DeflateConfiguration(8,16,128,128),new
zip_DeflateConfiguration(8,32,128,256),new zip_DeflateConfiguration(32,128,258,1024),new
zip_DeflateConfiguration(32,258,258,4096));function zip_DeflateCT(){this.fc=0,this.dl=0}function
zip_DeflateTreeDesc(){this.dyn_tree=null,this.static_tree=null,this.extra_bits=null,this.extra_base=0,this.
elems=0,this.max_length=0,this.max_code=0}function zip_DeflateConfiguration(_,i,p,z){this.good_length=_,this.
max_lazy=i,this.nice_length=p,this.max_chain=z}function zip_DeflateBuffer(){this.next=null,this.len=0,this.
ptr=new Array(zip_OUTBUFSIZ),this.off=0}function zip_deflate_start(_){var i;if(_?_<1?_=1:_>9&&(_=9):
_=zip_DEFAULT_LEVEL,zip_compr_level=_,zip_initflag=!1,zip_eofile=!1,null==zip_outbuf){for
(zip_free_queue=zip_qhead=zip_qtail=null,zip_outbuf=new Array(zip_OUTBUFSIZ),zip_window=new Array
(zip_window_size),zip_d_buf=new Array(zip_DIST_BUFSIZE),zip_l_buf=new Array(zip_INBUFSIZ+zip_INBUF_EXTRA),
zip_prev=new Array(1<<zip_BITS),zip_dyn_ltree=new Array(zip_HEAP_SIZE),i=0;i<zip_HEAP_SIZE;i++)zip_dyn_ltree[i]
=new zip_DeflateCT;for(zip_dyn_dtree=new Array(2*zip_D_CODES+1),i=0;i<2*zip_D_CODES+1;i++)zip_dyn_dtree[i]=new
zip_DeflateCT;for(zip_static_ltree=new Array(zip_L_CODES+2),i=0;i<zip_L_CODES+2;i++)zip_static_ltree[i]=new
zip_DeflateCT;for(zip_static_dtree=new Array(zip_D_CODES),i=0;i<zip_D_CODES;i++)zip_static_dtree[i]=new
zip_DeflateCT;for(zip_bl_tree=new Array(2*zip_BL_CODES+1),i=0;i<2*zip_BL_CODES+1;i++)zip_bl_tree[i]=new
zip_DeflateCT;zip_l_desc=new zip_DeflateTreeDesc,zip_d_desc=new zip_DeflateTreeDesc,zip_bl_desc=new
zip_DeflateTreeDesc,zip_bl_count=new Array(zip_MAX_BITS+1),zip_heap=new Array(2*zip_L_CODES+1),zip_depth=new
Array(2*zip_L_CODES+1),zip_length_code=new Array(zip_MAX_MATCH-zip_MIN_MATCH+1),zip_dist_code=new Array(512),
zip_base_length=new Array(zip_LENGTH_CODES),zip_base_dist=new Array(zip_D_CODES),zip_flag_buf=new Array(parseInt
(zip_LIT_BUFSIZE/8))}}function zip_deflate_end(){zip_free_queue=zip_qhead=zip_qtail=null,zip_outbuf=null,
zip_window=null,zip_d_buf=null,zip_l_buf=null,zip_prev=null,zip_dyn_ltree=null,zip_dyn_dtree=null,
zip_static_ltree=null,zip_static_dtree=null,zip_bl_tree=null,zip_l_desc=null,zip_d_desc=null,zip_bl_desc=null,
zip_bl_count=null,zip_heap=null,zip_depth=null,zip_length_code=null,zip_dist_code=null,zip_base_length=null,
zip_base_dist=null,zip_flag_buf=null}function zip_reuse_queue(_){_.next=zip_free_queue,zip_free_queue=_}
function zip_new_queue(){var _;return null!=zip_free_queue?(_=zip_free_queue,zip_free_queue=zip_free_queue.
next):_=new zip_DeflateBuffer,_.next=null,_.len=_.off=0,_}function zip_head1(_){return zip_prev[zip_WSIZE+_]}
function zip_head2(_,i){return zip_prev[zip_WSIZE+_]=i}function zip_put_byte(_){zip_outbuf
[zip_outoff+zip_outcnt++]=_,zip_outoff+zip_outcnt==zip_OUTBUFSIZ&&zip_qoutbuf()}function zip_put_short(_)
{_&=65535,zip_outoff+zip_outcnt<zip_OUTBUFSIZ-2?(zip_outbuf[zip_outoff+zip_outcnt++]=255&_,zip_outbuf
[zip_outoff+zip_outcnt++]=_>>>8):(zip_put_byte(255&_),zip_put_byte(_>>>8))}function zip_INSERT_STRING()
{zip_ins_h=(zip_ins_h<<zip_H_SHIFT^255&zip_window[zip_strstart+zip_MIN_MATCH-1])&zip_HASH_MASK,
zip_hash_head=zip_head1(zip_ins_h),zip_prev[zip_strstart&zip_WMASK]=zip_hash_head,zip_head2(zip_ins_h,
zip_strstart)}function zip_SEND_CODE(_,i){zip_send_bits(i[_].fc,i[_].dl)}function zip_D_CODE(_){return 255&
(_<256?zip_dist_code[_]:zip_dist_code[256+(_>>7)])}function zip_SMALLER(_,i,p){return _[i].fc<_[p].fc||_[i].
fc==_[p].fc&&zip_depth[i]<=zip_depth[p]}function zip_read_buff(_,i,p){var z;for(z=0;
z<p&&zip_deflate_pos<zip_deflate_data.length;z++)_[i+z]=255&zip_deflate_data.charCodeAt(zip_deflate_pos++);
return z}function zip_lm_init(){var _;for(_=0;_<zip_HASH_SIZE;_++)zip_prev[zip_WSIZE+_]=0;if
(zip_max_lazy_match=zip_configuration_table[zip_compr_level].max_lazy,zip_good_match=zip_configuration_table
[zip_compr_level].good_length,zip_FULL_SEARCH||(zip_nice_match=zip_configuration_table[zip_compr_level].
nice_length),zip_max_chain_length=zip_configuration_table[zip_compr_level].max_chain,zip_strstart=0,
zip_block_start=0,(zip_lookahead=zip_read_buff(zip_window,0,2*zip_WSIZE))<=0)return zip_eofile=!0,void
(zip_lookahead=0);for(zip_eofile=!1;zip_lookahead<zip_MIN_LOOKAHEAD&&!zip_eofile;)zip_fill_window();for
(zip_ins_h=0,_=0;_<zip_MIN_MATCH-1;_++)zip_ins_h=(zip_ins_h<<zip_H_SHIFT^255&zip_window[_])&zip_HASH_MASK}
function zip_longest_match(_){var i,p,z=zip_max_chain_length,e=zip_strstart,t=zip_prev_length,
a=zip_strstart>zip_MAX_DIST?zip_strstart-zip_MAX_DIST:zip_NIL,l=zip_strstart+zip_MAX_MATCH,n=zip_window[e+t-1],
r=zip_window[e+t];zip_prev_length>=zip_good_match&&(z>>=2);do{if(zip_window[(i=_)+t]==r&&zip_window[i+t-1]
==n&&zip_window[i]==zip_window[e]&&zip_window[++i]==zip_window[e+1]){e+=2,i++;do{}while(zip_window[++e]
==zip_window[++i]&&zip_window[++e]==zip_window[++i]&&zip_window[++e]==zip_window[++i]&&zip_window[++e]
==zip_window[++i]&&zip_window[++e]==zip_window[++i]&&zip_window[++e]==zip_window[++i]&&zip_window[++e]
==zip_window[++i]&&zip_window[++e]==zip_window[++i]&&e<l);if(p=zip_MAX_MATCH-(l-e),e=l-zip_MAX_MATCH,p>t){if
(zip_match_start=_,t=p,zip_FULL_SEARCH){if(p>=zip_MAX_MATCH)break}else if(p>=zip_nice_match)break;n=zip_window
[e+t-1],r=zip_window[e+t]}}}while((_=zip_prev[_&zip_WMASK])>a&&0!=--z);return t}function zip_fill_window(){var
_,i,p=zip_window_size-zip_lookahead-zip_strstart;if(-1==p)p--;else if(zip_strstart>=zip_WSIZE+zip_MAX_DIST){for
(_=0;_<zip_WSIZE;_++)zip_window[_]=zip_window[_+zip_WSIZE];for(zip_match_start-=zip_WSIZE,zip_strstart-
=zip_WSIZE,zip_block_start-=zip_WSIZE,_=0;_<zip_HASH_SIZE;_++)zip_head2(_,(i=zip_head1(_))>=zip_WSIZE?i-
zip_WSIZE:zip_NIL);for(_=0;_<zip_WSIZE;_++)i=zip_prev[_],zip_prev[_]=i>=zip_WSIZE?i-zip_WSIZE:zip_NIL;
p+=zip_WSIZE}zip_eofile||((_=zip_read_buff(zip_window,zip_strstart+zip_lookahead,p))<=0?zip_eofile=!0:
zip_lookahead+=_)}function zip_deflate_fast(){for(;0!=zip_lookahead&&null==zip_qhead;){var _;if
(zip_INSERT_STRING(),zip_hash_head!=zip_NIL&&zip_strstart-zip_hash_head<=zip_MAX_DIST&&
(zip_match_length=zip_longest_match(zip_hash_head))>zip_lookahead&&(zip_match_length=zip_lookahead),
zip_match_length>=zip_MIN_MATCH)if(_=zip_ct_tally(zip_strstart-zip_match_start,zip_match_length-zip_MIN_MATCH),
zip_lookahead-=zip_match_length,zip_match_length<=zip_max_lazy_match){zip_match_length--;do{zip_strstart++,
zip_INSERT_STRING()}while(0!=--zip_match_length);zip_strstart++}else zip_strstart+=zip_match_length,
zip_match_length=0,zip_ins_h=((zip_ins_h=255&zip_window[zip_strstart])<<zip_H_SHIFT^255&zip_window
[zip_strstart+1])&zip_HASH_MASK;else _=zip_ct_tally(0,255&zip_window[zip_strstart]),zip_lookahead--,
zip_strstart++;for(_&&(zip_flush_block(0),zip_block_start=zip_strstart);zip_lookahead<zip_MIN_LOOKAHEAD&&!
zip_eofile;)zip_fill_window()}}function zip_deflate_better(){for(;0!=zip_lookahead&&null==zip_qhead;){if
```

```
(zip_INSERT_STRING(),zip_prev_length=zip_match_length,zip_prev_match=zip_match_start,
zip_match_length=zip_MIN_MATCH-1,zip_hash_head!=zip_NIL&&zip_prev_length<zip_max_lazy_match&&zip_strstart-
zip_hash_head<=zip_MAX_DIST&&((zip_match_length=zip_longest_match(zip_hash_head))>zip_lookahead&&
(zip_match_length=zip_lookahead),zip_match_length==zip_MIN_MATCH&&zip_strstart-
zip_match_start>zip_TOO_FAR&&zip_match_length--),
zip_prev_length>=zip_MIN_MATCH&&zip_match_length<=zip_prev_length){var _;_=zip_ct_tally(zip_strstart-1-
zip_prev_match,zip_prev_length-zip_MIN_MATCH),zip_lookahead-=zip_prev_length-1,zip_prev_length-=2;do
{zip_strstart++,zip_INSERT_STRING()}while(0!=--zip_prev_length);zip_match_available=0,
zip_match_length=zip_MIN_MATCH-1,zip_strstart++,_&&(zip_flush_block(0),zip_block_start=zip_strstart)}else 0!
=zip_match_available?(zip_ct_tally(0,255&zip_window[zip_strstart-1])&&(zip_flush_block(0),
zip_block_start=zip_strstart),zip_strstart++,zip_lookahead--):(zip_match_available=1,zip_strstart++,
zip_lookahead--);for(;zip_lookahead<zip_MIN_LOOKAHEAD&&!zip_eofile;)zip_fill_window()}}function
zip_init_deflate(){zip_eofile||(zip_bi_buf=0,zip_bi_valid=0,zip_ct_init(),zip_lm_init(),zip_qhead=null,
zip_outcnt=0,zip_outoff=0,zip_compr_level<=3?(zip_prev_length=zip_MIN_MATCH-1,zip_match_length=0):
(zip_match_length=zip_MIN_MATCH-1,zip_match_available=0),zip_complete=!1)}function zip_deflate_internal(_,i,p)
{var z;return zip_initflag||(zip_init_deflate(),zip_initflag=!0,0!=zip_lookahead)?(z=zip_qcopy(_,i,p))==p?p:
zip_complete?z:(zip_compr_level<=3?zip_deflate_fast():zip_deflate_better(),0==zip_lookahead&&(0!
=zip_match_available&&zip_ct_tally(0,255&zip_window[zip_strstart-1]),zip_flush_block(1),zip_complete=!0),
z+zip_qcopy(_,z+i,p-z)):(zip_complete=!0,0)}function zip_qcopy(_,i,p){var z,e,t;for(z=0;null!=zip_qhead&&z<p;)
{for((e=p-z)>zip_qhead.len&&(e=zip_qhead.len),t=0;t<e;t++)_[i+z+t]=zip_qhead.ptr[zip_qhead.off+t];var a;if
(zip_qhead.off+=e,zip_qhead.len-=e,z+=e,0==zip_qhead.len)a=zip_qhead,zip_qhead=zip_qhead.next,zip_reuse_queue
(a)}if(z==p)return z;if(zip_outoff<zip_outcnt){for((e=p-z)>zip_outcnt-zip_outoff&&(e=zip_outcnt-zip_outoff),t=0;
t<e;t++)_[i+z+t]=zip_outbuf[zip_outoff+t];z+=e,zip_outcnt==(zip_outoff+=e)&&(zip_outcnt=zip_outoff=0)}return z}
function zip_ct_init(){var _,i,p,z,e;if(0==zip_static_dtree[0].dl){for(zip_l_desc.dyn_tree=zip_dyn_ltree,
zip_l_desc.static_tree=zip_static_ltree,zip_l_desc.extra_bits=zip_extra_lbits,zip_l_desc.
extra_base=zip_LITERALS+1,zip_l_desc.elems=zip_L_CODES,zip_l_desc.max_length=zip_MAX_BITS,zip_l_desc.max_code=0,
zip_d_desc.dyn_tree=zip_dyn_dtree,zip_d_desc.static_tree=zip_static_dtree,zip_d_desc.extra_bits=zip_extra_dbits,
zip_d_desc.extra_base=0,zip_d_desc.elems=zip_D_CODES,zip_d_desc.max_length=zip_MAX_BITS,zip_d_desc.max_code=0,
zip_bl_desc.dyn_tree=zip_bl_tree,zip_bl_desc.static_tree=null,zip_bl_desc.extra_bits=zip_extra_blbits,
zip_bl_desc.extra_base=0,zip_bl_desc.elems=zip_BL_CODES,zip_bl_desc.max_length=zip_MAX_BL_BITS,zip_bl_desc.
max_code=0,p=0,z=0;z<zip_LENGTH_CODES-1;z++)for(zip_base_length[z]=p,_=0;_<1<<zip_extra_lbits[z];_++)
zip_length_code[p++]=z;for(zip_length_code[p-1]=z,e=0,z=0;z<16;z++)for(zip_base_dist[z]=e,_=0;
_<1<<zip_extra_dbits[z];_++)zip_dist_code[e++]=z;for(e>>=7;z<zip_D_CODES;z++)for(zip_base_dist[z]=e<<7,_=0;
_<1<<zip_extra_dbits[z]-7;_++)zip_dist_code[256+e++]=z;for(i=0;i<=zip_MAX_BITS;i++)zip_bl_count[i]=0;for(_=0;
_<=143;)zip_static_ltree[_++].dl=8,zip_bl_count[8]++;for(;_<=255;)zip_static_ltree[_++].dl=9,zip_bl_count[9]++;
for(;_<=279;)zip_static_ltree[_++].dl=7,zip_bl_count[7]++;for(;_<=287;)zip_static_ltree[_++].dl=8,zip_bl_count
[8]++;for(zip_gen_codes(zip_static_ltree,zip_L_CODES+1),_=0;_<zip_D_CODES;_++)zip_static_dtree[_].dl=5,
zip_static_dtree[_].fc=zip_bi_reverse(_,5);zip_init_block()}}function zip_init_block(){var _;for(_=0;
_<zip_L_CODES;_++)zip_dyn_ltree[_].fc=0;for(_=0;_<zip_D_CODES;_++)zip_dyn_dtree[_].fc=0;for(_=0;_<zip_BL_CODES;
_++)zip_bl_tree[_].fc=0;zip_dyn_ltree[zip_END_BLOCK].fc=1,zip_opt_len=zip_static_len=0,
zip_last_lit=zip_last_dist=zip_last_flags=0,zip_flags=0,zip_flag_bit=1}function zip_pqdownheap(_,i){for(var
p=zip_heap[i],z=i<<1;z<=zip_heap_len&&(z<zip_heap_len&&zip_SMALLER(_,zip_heap[z+1],zip_heap[z])&&z++,!
zip_SMALLER(_,p,zip_heap[z]));)zip_heap[i]=zip_heap[z],i=z,z<<=1;zip_heap[i]=p}function zip_gen_bitlen(_){var i,
p,z,e,t,a,l=_.dyn_tree,n=_.extra_bits,r=_.extra_base,o=_.max_code,d=_.max_length,s=_.static_tree,f=0;for(e=0;
e<=zip_MAX_BITS;e++)zip_bl_count[e]=0;for(l[zip_heap[zip_heap_max]].dl=0,i=zip_heap_max+1;i<zip_HEAP_SIZE;i++)
(e=l[l[p=zip_heap[i]].dl].dl+1)>d&&(e=d,f++),l[p].dl=e,p>o||(zip_bl_count[e]++,t=0,p>=r&&(t=n[p-r]),a=l[p].fc,
zip_opt_len+=a*(e+t),null!=s&&(zip_static_len+=a*(s[p].dl+t)));if(0!=f){do{for(e=d-1;0==zip_bl_count[e];)e--;
zip_bl_count[e]--,zip_bl_count[e+1]+=2,zip_bl_count[d]--,f-=2}while(f>0);for(e=d;0!=e;e--)for(p=zip_bl_count[e];
0!=p;)(z=zip_heap[--i])>o||(l[z].dl!=e&&(zip_opt_len+=(e-l[z].dl)*l[z].fc,l[z].fc=e),p--)}}function
zip_gen_codes(_,i){var p,z,e=new Array(zip_MAX_BITS+1),t=0;for(p=1;p<=zip_MAX_BITS;p++)t=t+zip_bl_count[p-1]<<1,
e[p]=t;for(z=0;z<=i;z++){var a=_[z].dl;0!=a&&(_[z].fc=zip_bi_reverse(e[a]++,a))}}function zip_build_tree(_){var
i,p,z=_.dyn_tree,e=_.static_tree,t=_.elems,a=-1,l=t;for(zip_heap_len=0,zip_heap_max=zip_HEAP_SIZE,i=0;i<t;i++)0!
=z[i].fc?(zip_heap[++zip_heap_len]=a=i,zip_depth[i]=0):z[i].dl=0;for(;zip_heap_len<2;){var n=zip_heap
[++zip_heap_len]=a<2?++a:0;z[n].fc=1,zip_depth[n]=0,zip_opt_len--,null!=e&&(zip_static_len-=e[n].dl)}for(_.
max_code=a,i=zip_heap_len>>1;i>=1;i--)zip_pqdownheap(z,i);do{i=zip_heap[zip_SMALLEST],zip_heap[zip_SMALLEST]
=zip_heap[zip_heap_len--],zip_pqdownheap(z,zip_SMALLEST),p=zip_heap[zip_SMALLEST],zip_heap[--zip_heap_max]=i,
zip_heap[--zip_heap_max]=p,z[l].fc=z[i].fc+z[p].fc,zip_depth[i]>zip_depth[p]+1?zip_depth[l]=zip_depth[i]:
zip_depth[l]=zip_depth[p]+1,z[i].dl=z[p].dl=l,zip_heap[zip_SMALLEST]=l++,zip_pqdownheap(z,zip_SMALLEST)}while
(zip_heap_len>=2);zip_heap[--zip_heap_max]=zip_heap[zip_SMALLEST],zip_gen_bitlen(_),zip_gen_codes(z,a)}function
zip_scan_tree(_,i){var p,z,e=-1,t=_[0].dl,a=0,l=7,n=4;for(0==t&&(l=138,n=3),_[i+1].dl=65535,p=0;p<=i;p++)z=t,t=_
[p+1].dl,++a<l&&z==t||(a<n?zip_bl_tree[z].fc+=a:0!=z?(z!=e&&zip_bl_tree[z].fc++,zip_bl_tree[zip_REP_3_6].fc++):
a<=10?zip_bl_tree[zip_REPZ_3_10].fc++:zip_bl_tree[zip_REPZ_11_138].fc++,a=0,e=z,0==t?(l=138,n=3):z==t?(l=6,n=3):
(l=7,n=4))}function zip_send_tree(_,i){var p,z,e=-1,t=_[0].dl,a=0,l=7,n=4;for(0==t&&(l=138,n=3),p=0;p<=i;p++)if
(z=t,t=_[p+1].dl,!(++a<l&&z==t)){if(a<n)do{zip_SEND_CODE(z,zip_bl_tree)}while(0!=--a);else 0!=z?(z!=e&&
(zip_SEND_CODE(z,zip_bl_tree),a--),zip_SEND_CODE(zip_REP_3_6,zip_bl_tree),zip_send_bits(a-3,2)):a<=10?
(zip_SEND_CODE(zip_REPZ_3_10,zip_bl_tree),zip_send_bits(a-3,3)):(zip_SEND_CODE(zip_REPZ_11_138,zip_bl_tree),
zip_send_bits(a-11,7));a=0,e=z,0==t?(l=138,n=3):z==t?(l=6,n=3):(l=7,n=4)}}function zip_build_bl_tree(){var _;for
(zip_scan_tree(zip_dyn_ltree,zip_l_desc.max_code),zip_scan_tree(zip_dyn_dtree,zip_d_desc.max_code),
zip_build_tree(zip_bl_desc),_=zip_BL_CODES-1;_>=3&&0==zip_bl_tree[zip_bl_order[_]].dl;_--);return
zip_opt_len+=3*(_+1)+5+5+4,_}function zip_send_all_trees(_,i,p){var z;for(zip_send_bits(_-257,5),zip_send_bits
(i-1,5),zip_send_bits(p-4,4),z=0;z<p;z++)zip_send_bits(zip_bl_tree[zip_bl_order[z]].dl,3);zip_send_tree
```

```
(zip_dyn_ltree,_-1),zip_send_tree(zip_dyn_dtree,i-1)}function zip_flush_block(_){var i,p,z,e,t;if
(e=zip_strstart-zip_block_start,zip_flag_buf[zip_last_flags]=zip_flags,zip_build_tree(zip_l_desc),zip_build_tree
(zip_d_desc),z=zip_build_bl_tree(),(p=zip_static_len+3+7>>3)<=(i=zip_opt_len+3+7>>3)&&(i=p),
e+4<=i&&zip_block_start>=0)for(zip_send_bits((zip_STORED_BLOCK<<1)+_,3),zip_bi_windup(),zip_put_short(e),
zip_put_short(~e),t=0;t<e;t++)zip_put_byte(zip_window[zip_block_start+t]);else p==i?(zip_send_bits
((zip_STATIC_TREES<<1)+_,3),zip_compress_block(zip_static_ltree,zip_static_dtree)):(zip_send_bits
((zip_DYN_TREES<<1)+_,3),zip_send_all_trees(zip_l_desc.max_code+1,zip_d_desc.max_code+1,z+1),zip_compress_block
(zip_dyn_ltree,zip_dyn_dtree));zip_init_block(),0!=_&&zip_bi_windup()}function zip_ct_tally(_,i){if(zip_l_buf
[zip_last_lit++]=i,0==_?zip_dyn_ltree[i].fc++:(_--,zip_dyn_ltree[zip_length_code[i]+zip_LITERALS+1].fc++,
zip_dyn_dtree[zip_D_CODE(_)].fc++,zip_d_buf[zip_last_dist++]=_,zip_flags|=zip_flag_bit),zip_flag_bit<<=1,0==
(7&zip_last_lit)&&(zip_flag_buf[zip_last_flags++]=zip_flags,zip_flags=0,zip_flag_bit=1),zip_compr_level>2&&0==
(4095&zip_last_lit)){var p,z=8*zip_last_lit,e=zip_strstart-zip_block_start;for(p=0;p<zip_D_CODES;p++)
z+=zip_dyn_dtree[p].fc*(5+zip_extra_dbits[p]);if(z>>=3,zip_last_dist<parseInt(zip_last_lit/2)&&z<parseInt(e/2))
return!0}return zip_last_lit==zip_LIT_BUFSIZE-1||zip_last_dist==zip_DIST_BUFSIZE}function zip_compress_block(_,
i){var p,z,e,t,a=0,l=0,n=0,r=0;if(0!=zip_last_lit)do{0==(7&a)&&(r=zip_flag_buf[n++]),z=255&zip_l_buf[a++],0==
(1&r)?zip_SEND_CODE(z,_):(zip_SEND_CODE((e=zip_length_code[z])+zip_LITERALS+1,_),0!=(t=zip_extra_lbits[e])
&&zip_send_bits(z-=zip_base_length[e],t),zip_SEND_CODE(e=zip_D_CODE(p=zip_d_buf[l++]),i),0!=(t=zip_extra_dbits
[e])&&zip_send_bits(p-=zip_base_dist[e],t)),r>>=1}while(a<zip_last_lit);zip_SEND_CODE(zip_END_BLOCK,_)}var
zip_Buf_size=16;function zip_send_bits(_,i){zip_bi_valid>zip_Buf_size-i?(zip_put_short
(zip_bi_buf|=_<<zip_bi_valid),zip_bi_buf=_>>zip_Buf_size-zip_bi_valid,zip_bi_valid+=i-zip_Buf_size):
(zip_bi_buf|=_<<zip_bi_valid,zip_bi_valid+=i)}function zip_bi_reverse(_,i){var p=0;do{p|=1&_,_>>=1,p<<=1}while
(--i>0);return p>>1}function zip_bi_windup(){zip_bi_valid>8?zip_put_short(zip_bi_buf):
zip_bi_valid>0&&zip_put_byte(zip_bi_buf),zip_bi_buf=0,zip_bi_valid=0}function zip_qoutbuf(){if(0!=zip_outcnt)
{var _,i;for(_=zip_new_queue(),null==zip_qhead?zip_qhead=zip_qtail=_:zip_qtail=zip_qtail.next=_,_.
len=zip_outcnt-zip_outoff,i=0;i<_.len;i++)_.ptr[i]=zip_outbuf[zip_outoff+i];zip_outcnt=zip_outoff=0}}function
zip_deflate(_,i){var p,z,e,t;for(zip_deflate_data=_,zip_deflate_pos=0,void 0===i&&(i=zip_DEFAULT_LEVEL),
zip_deflate_start(i),z=new Array(1024),p="";(e=zip_deflate_internal(z,0,z.length))>0;)for(t=0;t<e;t++)p+=String.
fromCharCode(z[t]);return zip_deflate_data=null,p}
var zip_slide,zip_wp,zip_fixed_td,zip_fixed_bl,fixed_bd,zip_bit_buf,zip_bit_len,zip_method,zip_eof,
zip_copy_leng,zip_copy_dist,zip_tl,zip_td,zip_bl,zip_bd,zip_inflate_data,zip_inflate_pos,zip_WSIZE=32768,
zip_STORED_BLOCK=0,zip_STATIC_TREES=1,zip_DYN_TREES=2,zip_lbits=9,zip_dbits=6,zip_INBUFSIZ=32768,
zip_INBUF_EXTRA=64,zip_fixed_tl=null,zip_MASK_BITS=new Array
(0,1,3,7,15,31,63,127,255,511,1023,2047,4095,8191,16383,32767,65535),zip_cplens=new Array
(3,4,5,6,7,8,9,10,11,13,15,17,19,23,27,31,35,43,51,59,67,83,99,115,131,163,195,227,258,0,0),zip_cplext=new Array
(0,0,0,0,0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,0,99,99),zip_cpdist=new Array
(1,2,3,4,5,7,9,13,17,25,33,49,65,97,129,193,257,385,513,769,1025,1537,2049,3073,4097,6145,8193,12289,16385,24577
),zip_cpdext=new Array(0,0,0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,13,13),zip_border=new Array
(16,17,18,0,8,7,9,6,10,5,11,4,12,3,13,2,14,1,15);function zip_HuftList(){this.next=null,this.list=null}function
zip_HuftNode(){this.e=0,this.b=0,this.n=0,this.t=null}function zip_HuftBuild(i,_,p,z,t,e){this.BMAX=16,this.
N_MAX=288,this.status=0,this.root=null,this.m=0;var n,r,l,f,o,s,d,u,a,T,B,S,E,I,b,c,h,D=new Array(this.BMAX+1),
w=new Array(this.BMAX+1),y=new zip_HuftNode,M=new Array(this.BMAX),A=new Array(this.N_MAX),N=new Array(this.
BMAX+1);for(h=this.root=null,s=0;s<D.length;s++)D[s]=0;for(s=0;s<w.length;s++)w[s]=0;for(s=0;s<M.length;s++)M[s]
=null;for(s=0;s<A.length;s++)A[s]=0;for(s=0;s<N.length;s++)N[s]=0;r=_>256?i[256]:this.BMAX,a=i,T=0,s=_;do{D[a
[T]]++,T++}while(--s>0);if(D[0]==_)return this.root=null,this.m=0,void(this.status=0);for(d=1;d<=this.BMAX&&0==D
[d];d++);for(u=d,e<d&&(e=d),s=this.BMAX;0!=s&&0==D[s];s--);for(f=s,e>s&&(e=s),I=1<<d;d<s;d++,I<<=1)if((I-=D[d])
<0)return this.status=2,void(this.m=e);if((I-=D[s])<0)return this.status=2,void(this.m=e);for(D[s]+=I,N[1]=d=0,
a=D,T=1,E=2;--s>0;)N[E++]=d+=a[T++];a=i,T=0,s=0;do{0!=(d=a[T++])&&(A[N[d]++]=s)}while(++s<_);for(_=N[f],N[0]
=s=0,a=A,T=0,o=-1,S=w[0]=0,B=null,b=0;u<=f;u++)for(n=D[u];n-- >0;){for(;u>S+w[1+o];){if(S+=w[1+o],o++,b=(b=f-S)
>e?e:b,(l=1<<(d=u-S))>n+1)for(l-=n+1,E=u;++d<b&&!((l<<=1)<=D[++E]);)l-=D[E];for(S+d>r&&S<r&&(d=r-S),b=1<<d,w
[1+o]=d,B=new Array(b),c=0;c<b;c++)B[c]=new zip_HuftNode;(h=null==h?this.root=new zip_HuftList:h.next=new
zip_HuftList).next=null,h.list=B,M[o]=B,o>0&&(N[o]=s,y.b=w[o],y.e=16+d,y.t=B,d=(s&(1<<S)-1)>>S-w[o],M[o-1][d].
e=y.e,M[o-1][d].b=y.b,M[o-1][d].n=y.n,M[o-1][d].t=y.t)}for(y.b=u-S,T>=_?y.e=99:a[T]<p?(y.e=a[T]<256?16:15,y.n=a
[T++]):(y.e=t[a[T]-p],y.n=z[a[T++]-p]),l=1<<u-S,d=s>>S;d<b;d+=l)B[d].e=y.e,B[d].b=y.b,B[d].n=y.n,B[d].t=y.t;for
(d=1<<u-1;0!=(s&d);d>>=1)s^=d;for(s^=d;(s&(1<<S)-1)!=N[o];)S-=w[o],o--}this.m=w[1],this.status=0!=I&&1!=f?1:0}
function zip_GET_BYTE(){return zip_inflate_data.length==zip_inflate_pos?-1:255&zip_inflate_data.charCodeAt
(zip_inflate_pos++)}function zip_NEEDBITS(i){for(;zip_bit_len<i;)zip_bit_buf|=zip_GET_BYTE()<<zip_bit_len,
zip_bit_len+=8}function zip_GETBITS(i){return zip_bit_buf&zip_MASK_BITS[i]}function zip_DUMPBITS(i)
{zip_bit_buf>>=i,zip_bit_len-=i}function zip_inflate_codes(i,_,p){var z,t,e;if(0==p)return 0;for(e=0;;){for
(zip_NEEDBITS(zip_bl),z=(t=zip_tl.list[zip_GETBITS(zip_bl)]).e;z>16;){if(99==z)return-1;zip_DUMPBITS(t.b),
zip_NEEDBITS(z-=16),z=(t=t.t[zip_GETBITS(z)]).e}if(zip_DUMPBITS(t.b),16!=z){if(15==z)break;for(zip_NEEDBITS(z),
zip_copy_leng=t.n+zip_GETBITS(z),zip_DUMPBITS(z),zip_NEEDBITS(zip_bd),z=(t=zip_td.list[zip_GETBITS(zip_bd)]).e;
z>16;){if(99==z)return-1;zip_DUMPBITS(t.b),zip_NEEDBITS(z-=16),z=(t=t.t[zip_GETBITS(z)]).e}for(zip_DUMPBITS(t.
b),zip_NEEDBITS(z),zip_copy_dist=zip_wp-t.n-zip_GETBITS(z),zip_DUMPBITS(z);zip_copy_leng>0&&e<p;)zip_copy_leng--
,zip_copy_dist&=zip_WSIZE-1,zip_wp&=zip_WSIZE-1,i[_+e++]=zip_slide[zip_wp++]=zip_slide[zip_copy_dist++];if(e==p)
return p}else if(zip_wp&=zip_WSIZE-1,i[_+e++]=zip_slide[zip_wp++]=t.n,e==p)return p}return zip_method=-1,e}
function zip_inflate_stored(i,_,p){var z;if(zip_DUMPBITS(z=7&zip_bit_len),zip_NEEDBITS(16),z=zip_GETBITS(16),
zip_DUMPBITS(16),zip_NEEDBITS(16),z!=(65535&~zip_bit_buf))return-1;for(zip_DUMPBITS(16),zip_copy_leng=z,z=0;
zip_copy_leng>0&&z<p;)zip_copy_leng--,zip_wp&=zip_WSIZE-1,zip_NEEDBITS(8),i[_+z++]=zip_slide[zip_wp++]
=zip_GETBITS(8),zip_DUMPBITS(8);return 0==zip_copy_leng&&(zip_method=-1),z}function zip_inflate_fixed(i,_,p){if
(null==zip_fixed_tl){var z,t,e=new Array(288);for(z=0;z<144;z++)e[z]=8;for(;z<256;z++)e[z]=9;for(;z<280;z++)e[z]
=7;for(;z<288;z++)e[z]=8;if(0!=(t=new zip_HuftBuild(e,288,257,zip_cplens,zip_cplext,zip_fixed_bl=7)).status)
```

```
return alert("HufBuild error: "+t.status),-1;for(zip_fixed_tl=t.root,zip_fixed_bl=t.m,z=0;z<30;z++)e[z]=5;if
(zip_fixed_bd=5,(t=new zip_HuftBuild(e,30,0,zip_cpdist,zip_cpdext,zip_fixed_bd)).status>1)return
zip_fixed_tl=null,alert("HufBuild error: "+t.status),-1;zip_fixed_td=t.root,zip_fixed_bd=t.m}return
zip_tl=zip_fixed_tl,zip_td=zip_fixed_td,zip_bl=zip_fixed_bl,zip_bd=zip_fixed_bd,zip_inflate_codes(i,_,p)}
function zip_inflate_dynamic(i,_,p){var z,t,e,n,r,l,f,o,s,d=new Array(316);for(z=0;z<d.length;z++)d[z]=0;if
(zip_NEEDBITS(5),f=257+zip_GETBITS(5),zip_DUMPBITS(5),zip_NEEDBITS(5),o=1+zip_GETBITS(5),zip_DUMPBITS(5),
zip_NEEDBITS(4),l=4+zip_GETBITS(4),zip_DUMPBITS(4),f>286||o>30)return-1;for(t=0;t<l;t++)zip_NEEDBITS(3),d
[zip_border[t]]=zip_GETBITS(3),zip_DUMPBITS(3);for(;t<19;t++)d[zip_border[t]]=0;if(0!=(s=new zip_HuftBuild(d,
19,19,null,null,zip_bl=7)).status)return-1;for(zip_tl=s.root,zip_bl=s.m,n=f+o,z=e=0;z<n;)if(zip_NEEDBITS
(zip_bl),zip_DUMPBITS(t=(r=zip_tl.list[zip_GETBITS(zip_bl)]).b),(t=r.n)<16)d[z++]=e=t;else if(16==t){if
(zip_NEEDBITS(2),t=3+zip_GETBITS(2),zip_DUMPBITS(2),z+t>n)return-1;for(;t-- >0;)d[z++]=e}else if(17==t){if
(zip_NEEDBITS(3),t=3+zip_GETBITS(3),zip_DUMPBITS(3),z+t>n)return-1;for(;t-- >0;)d[z++]=0;e=0}else{if
(zip_NEEDBITS(7),t=11+zip_GETBITS(7),zip_DUMPBITS(7),z+t>n)return-1;for(;t-- >0;)d[z++]=0;e=0}if(s=new
zip_HuftBuild(d,f,257,zip_cplens,zip_cplext,zip_bl=zip_lbits),0==zip_bl&&(s.status=1),0!=s.status)return s.
status,-1;for(zip_tl=s.root,zip_bl=s.m,z=0;z<o;z++)d[z]=d[z+f];return s=new zip_HuftBuild(d,o,0,zip_cpdist,
zip_cpdext,zip_bd=zip_dbits),zip_td=s.root,0==(zip_bd=s.m)&&f>257?-1:(s.status,0!=s.status?-1:zip_inflate_codes
(i,_,p))}function zip_inflate_start(){null==zip_slide&&(zip_slide=new Array(2*zip_WSIZE)),zip_wp=0,
zip_bit_buf=0,zip_bit_len=0,zip_method=-1,zip_eof=!1,zip_copy_leng=zip_copy_dist=0,zip_tl=null}function
zip_inflate_internal(i,_,p){var z,t;for(z=0;z<p;){if(zip_eof&&-1==zip_method)return z;if(zip_copy_leng>0){if
(zip_method!=zip_STORED_BLOCK)for(;zip_copy_leng>0&&z<p;)zip_copy_leng--,zip_copy_dist&=zip_WSIZE-1,
zip_wp&=zip_WSIZE-1,i[_+z++]=zip_slide[zip_wp++]=zip_slide[zip_copy_dist++];else{for(;zip_copy_leng>0&&z<p;)
zip_copy_leng--,zip_wp&=zip_WSIZE-1,zip_NEEDBITS(8),i[_+z++]=zip_slide[zip_wp++]=zip_GETBITS(8),zip_DUMPBITS(8);
0==zip_copy_leng&&(zip_method=-1)}if(z==p)return z}if(-1==zip_method){if(zip_eof)break;zip_NEEDBITS(1),0!
=zip_GETBITS(1)&&(zip_eof=!0),zip_DUMPBITS(1),zip_NEEDBITS(2),zip_method=zip_GETBITS(2),zip_DUMPBITS(2),
zip_tl=null,zip_copy_leng=0}switch(zip_method){case 0:t=zip_inflate_stored(i,_+z,p-z);break;case 1:t=null!
=zip_tl?zip_inflate_codes(i,_+z,p-z):zip_inflate_fixed(i,_+z,p-z);break;case 2:t=null!=zip_tl?zip_inflate_codes
(i,_+z,p-z):zip_inflate_dynamic(i,_+z,p-z);break;default:t=-1}if(-1==t)return zip_eof?0:-1;z+=t}return z}
function zip_inflate(i){var _,p,z,t;zip_inflate_start(),zip_inflate_data=i,zip_inflate_pos=0;var e=-1;for(p=new
Array(1024),_="";(z=zip_inflate_internal(p,0,p.length))>0&&e!=zip_inflate_pos;)for(e=zip_inflate_pos,t=0;t<z;
t++)_+=String.fromCharCode(p[t]);return zip_inflate_data=null,_}
```
</script>

<!--
from http://www.broofa.com
mailto:robert@broofa.com
-->
<script>
```
function randomGUID(){return'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g,function(c){var r=Math.
random()*16|0,v=c=='x'?r:(r&0x3|0x8);return v.toString(16)})}
```
</script>

<!-- vkbeautify (modified)
from https://code.google.com/archive/p/vkbeautify/
 -->
<script>
```
function createShiftArr(e){var a="    ";if(isNaN(parseInt(e)))a=e;else switch(e){case 1:a=" ";break;case 2:a="
";break;case 3:a="   ";break;case 4:a="    ";break;case 5:a="     ";break;case 6:a="      ";break;case 7:
a="       ";break;case 8:a="        ";break;case 9:a="         ";break;case 10:a="          ";break;case 11:
a="           ";break;case 12:a="            "}var s=["\n"];for(ix=0;ix<100;ix++)s.push(s[ix]+a);return s}
function vkbeautify(){this.step="    ",this.shift=createShiftArr(this.step),this.xml=vkbeautifyxml}function
vkbeautifyxml(e,a){var s=e.replace(/\s+</g,"<").replace(/</g,"~::~<").replace(/\s*xmlns\:/g,"~::~xmlns:").
replace(/\s*xmlns\=/g,"~::~xmlns=").replace(/\s*([^\s]*\=)/g,"~::~$1").split("~::~"),r=s.length,c=!1,h=0,t="",
i=0,l=a?createShiftArr(a):this.shift;for(i=0;i<r;i++)s[i].search(/<!/)>-1?(t+=l[h]+s[i],c=!0,(s[i].search(/-->/)
>-1||s[i].search(/\]>/)>-1||s[i].search(/!DOCTYPE/)>-1)&&(c=!1)):s[i].search(/-->/)>-1||s[i].search(/\]>/)>-1?
(t+=s[i],c=!1):/^<\w/.exec(s[i-1])&&/^<\/\w/.exec(s[i])&&/^<[\w:\-\.\,]+/.exec(s[i-1])==/^<\/[\w:\-\.\,]+/.exec
(s[i])[0].replace("/","")?(t+=s[i],c||h--):s[i].search(/<\w/)>-1&&-1==s[i].search(/<\//)&&-1==s[i].search(/\/>
/)?t=t+=c?s[i]:l[h++]+s[i]:s[i].search(/<\w/)>-1&&s[i].search(/<\//)>-1?t=t+=c?s[i]:l[h]+s[i]:s[i].search(/<\//)
>-1?t=t+=c?s[i]:l[--h]+s[i]:s[i].search(/[^\s]*\=[^\s]*\s*\/>/)>-1?t=t+=c?s[i]:l[h--]+s[i]:s[i].search(/\/>/)>-
1?t=t+=c?s[i]:l[h]+s[i]:s[i].search(/<\?/)>-1?t+=l[h]+s[i]:s[i].search(/[^\s]*\=/)>-1?t+=l[h]+s[i]:t+=s[i];
return"\n"==t[0]?t.slice(1):t}beautify=new vkbeautify;
```
</script>

^{by Dan Malone <dmalone@calpoly.edu>}
</body>
</html>