

- [機能拡張について](#)
- [機能拡張サンプル](#)
 - [ニコフィルターをNicoPlayer置換フィルタへ変換して表示する \(-ver0.49\)](#)
 - [動画ページを全て表示する](#)
 - [過去のコメントを取得する \(-ver0.42\)](#)
 - [コメントをこっそり投稿する \(-ver0.45\)](#)

機能拡張について

[動画情報ウィンドウ#機能拡張](#)をご覧ください。

[動画情報ウィンドウ 機能拡張サンプル](#)

が最新版、こちらは過去ログです。
ページ容量制限で引っかかったら需要がなさそうなものから移してください。

機能拡張サンプル

ニコフィルターをNicoPlayer置換フィルタへ変換して表示する (-ver0.49)

表示するだけです。環境に合わせて[LastReplace]などが設定されている有効なフィルタに追記してください。

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript">
window.onload = function() {
var http = new ActiveXObject( "Msxml2.XMLHTTP" );
http.open( "GET", "http://www.nicovideo.jp/api/getflv?v=%_MOVIE_NAME_" , true );
http.onreadystatechange = function() {
if( http.readyState == 4 && http.status == 200 ) { onpage( http ); }
}
http.send();
}

UnescapeUTF8=function(str){
return str.replace(/%(E(0%[AB])|[1-CEF%][89AB]|D%[89])[0-9A-F]|C[2-9A-F]|D[0-9A-F])%[89AB][0-9A-F]|%[0-7][0-9A-F]/ig, fu
: ((c&15)<<6|parseInt(s.substring(4),16)&63)<<6|parseInt(s.substring(7),16)&63)
});
}

onpage = function( response ) {
if( response.responseText.match( /ng_up=(.*)/ ) ) {
var list = UnescapeUTF8( UnescapeUTF8( RegExp.$1 ) ).split( "&" );
for( var i = 0; i < list.length; i++ ) {
var pair = list[i].split( "=" );
display.innerHTML
+= "(/>(" + pair[0] + ")</) = (> + pair[1] + "<)<br />"
}
}
}
</script>
</head>
<body>
<div id="display"></div>
</body>
</html>

```

動画ページを全て表示する

[こちら](#)を参考に必要部分を抽出。テキストデータは全てdispTxtに入るので加工用リソースにどうぞ。

```

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=shift_jis">
<script language="javascript">
<!--
//XMLHttpRequestオブジェクト生成
function createHttpRequest()
{
if( window.ActiveXObject )

```

```

{
  try
  {
    return new ActiveXObject( "Msxml2.XMLHTTP" )
  } catch ( e ) {
    try
    {
      return new ActiveXObject( "Microsoft.XMLHTTP" )
    } catch ( e2 ) {
      return null
    }
  }
} else if( window.XMLHttpRequest )
{
  return new XMLHttpRequest()
} else {
  return null
}
}

```

```

//受信内容を確認してイベントを起動
function getFile( data , method , fileName , async )
{

```

```

  //XMLHttpRequestオブジェクト生成
  var httpoj = createHttpRequest()
  httpoj.open( method , fileName , async )

```

```

  //受信時に起動するイベント
  httpoj.onreadystatechange = function()
  {
    if ( httpoj.readyState==4 )
    {
      //コールバック
      on_loaded( httpoj )
    }
  }
  httpoj.send( data )
}

```

```

//コールバック関数(受信時に実行)
function on_loaded( oj )
{

```

```

  dispTxt = oj.responseText
  document.write( dispTxt );
}

```

```

//-->
</script>
</head>

```

```

<body onLoad="getFile( '', 'GET', 'http://www.nicovideo.jp/watch/sm%__MOVIE_ID__', true )" >
</body>
</html>

```

過去のコメントを取得する (-ver0.42)

指定日時のコメントを取得します。日時指定が泣けるほど手抜きです。過去のコメントはview_counterがないのですが、そのままだとNicoPlayerが飛んでしまうので暫定的に値0で要素を追加しました。

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
  function execBuildupCommand( argument )
  {
    var shellObject = new ActiveXObject( "WScript.Shell" );
    shellObject.Run( " \ "%_SOFT_DIR2_% \ NicoPlayer.exe \ " " + argument, 0, true );
    shellObject = null;
  }

  function getComment()
  {
    var vid = "%_MOVIE_NAME_%";
    var net = new ActiveXObject( "Msxml2.XMLHTTP" );

    // 動画ページへアクセス(APIを使うために必要らしい)
    net.open( "GET", "http://www.nicovideo.jp/watch/" + vid, false );

```

```

net.send();

// ダウンロード元サーバ・スレッド番号・ユーザIDを取得
var parametersHash = $H( {} );
net.open( "GET", "http://www.nicovideo.jp/api/getflv?v=" + vid, false );
net.onreadystatechange = function() {
  if( net.readyState == 4 && net.status == 200 ) {
    var parametersList = $( unescape( net.responseText ).split( "&" ) );
    parametersList.each( function( parameter ) {
      if( parameter.match( /^[^=]+=(.+)/ ) ) {
        parametersHash[RegExp.$1] = RegExp.$2;
      }
    } );
  }
}
net.send();

// キー取得(過去コメント取得に必要ならしい)
var waybackkey = "";
net.open( "GET", "http://www.nicovideo.jp/api/getwaybackkey?thread=" + parametersHash["thread_id"], false );
net.onreadystatechange = function() {
  if( net.readyState == 4 && net.status == 200 ) {
    if( net.responseText.match( /waybackkey=(.+)/ ) ) {
      waybackkey = RegExp.$1;
    }
  }
}
net.send();

// コメント取得
net.open( "POST", parametersHash["ms"], false );
net.onreadystatechange = function() {
  if( net.readyState == 4 && net.status == 200 ) {
    var xml = new ActiveXObject( "Microsoft.XMLDOM" );
    xml.loadXML( net.responseText );
    var chats = $( xml.getElementsByTagName( "chat" ) );

    // view_counterを追加(暫定)
    var viewCounterNode = xml.createElement( "view_counter" );
    viewCounterNode.setAttribute( "video", 0 );
    xml.getElementsByTagName( "packet" )[0].appendChild( viewCounterNode );

    // ファイルに保存
    var newCommentPath = "%_DOWNLOAD_DIR2_% \ \ %_TITLE_%(%_MOVIE_NAME_%)[ " + $( "date" ) + " ].xml";
    xml.save( newCommentPath );
    xml = null;

    // プレイリストに追加
    execBuildupCommand( " \ " + newCommentPath + " \ " -addlist " );

    $( "target" ).innerHTML = "End";
  }
}
var times = $( "date" ).split( "_" );
var time = Date.UTC( times[0], times[1] - 1, times[2], times[3], times[4] );
var when = ( time - 9 * 60 * 60 * 1000 ) / 1000;
net.send( "<thread res_from= \ "-1000 \ "
  + " when= \ " + when + " \ "
  + " waybackkey= \ " + waybackkey + " \ "
  + " user_id= \ " + parametersHash["user_id"] + " \ "
  + " version= \ "20061206 \ "
  + " thread= \ " + parametersHash["thread_id"] + " \ "
  + "/>"
);
net = null;
}
</script>
</head>
<body>
  日付<input id="date" type="text" value="2007_12_08_21_00"></input>
  <button onclick="getComment()">Get</button><br/>
  <span id="target"></span>
</body>
</html>

```

コメントをこっそり投稿する (-ver0.45)

自分専用コメントファイルを作り、そこに投稿します。他の誰にも見えません。コメ職人気分が味わえるかもしれません。再生時間は1/100秒単位で指定してください。

```
<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
var $_ = function( target, key ) {
    return $( target.getElementsByTagName( key ) );
}

// ローカルコメントファイル作成
var createTrialCommentFile = function( path ) {
    var fs = new ActiveXObject( "Scripting.FileSystemObject" );
    if( fs.FileExists( path ) ) return;
    var file = fs.CreateTextFile( path );
    file.Write( "<?xml version=\ "1.0\ " encoding=\ "UTF-8\ "?></packet>" );
    file.Close();
}

// ローカルコメントファイルにコメント追加
var xml;
var postCommentTrial = function() {
    var path = "%_DOWNLOAD_DIR2_% \ \%_TITLE_%(%_MOVIE_NAME_%)[Local].xml";
    createTrialCommentFile( path );
    if( !xml ) {
        xml = new ActiveXObject( "Microsoft.XMLDOM" );
        xml.async = false;
    }
    xml.load( path );
    var packet = $_( xml, "packet" )[0];
    var chat = xml.createElement( "chat" );
    chat.setAttribute( "vpos", $( "time" ) );
    chat.setAttribute( "mail", $( "command" ) );
    chat.text = ( $( "comment" ) );
    packet.appendChild( chat );
    xml.save( path );
}

var execBuildupCommand = function( param ) {
    var shell = new ActiveXObject( "WScript.Shell" );
    shell.Run( " \ \%_SOFT_DIR2_% \ \NicoPlayer.exe \ " + param, 0, true );
    shell = null;
}
</script>
</head>
<body>
<input id="time" value="0" />[1/100(s)]
<input id="command" value="ue" />
<input id="comment" value="test" />
<button onclick="postCommentTrial()">投稿</button>
<div id="target"></div>
</body>
</html>
```