

```

/*
    This is a procedure, not an udf because of the exec insert at the end
    The exec insert is needed because the response can exceed 8000 characters

    Normally, one should call methods like this :
    exec      @rc      = master.dbo.sp_OAMethod @http, 'Open', NULL, 'POST' , 'www.d-
trix.com/default.asp', 0

    But this does not work with WinHttp.WinHttpRequest.5.1 and gives different
errors
    So we call methods like this
    exec      @rc      = master.dbo.sp_OAMethod @http, 'Open("POST","www.d-trix.com/
default.asp",0)'

    if a (temp) table with the name #usp_httppost exists, the procedure will
insert the result into that table.
    if not, the procedure will return a resultset with the results
*/
create procedure dbo.usp_httppost
(
    @URL varchar(512),
    @post varchar(4000),
    @WebLogin varchar(128) = null,
    @WebPassword varchar(128) = null,
    @ProxyLogin varchar(128) = null,
    @ProxyPassword varchar(128) = null
)
/*
returns @usp_httppost table
(
    return_status bit null, -- 0 is OK, 1 is error
    error_msg varchar(4096) null,
    HTTP_Status varchar(30) null,
    HTTP_AllResponseHeaders text null,
    HTTP_ResponseText text null
)
*/
as
begin
    /*****
Name          : usp_httppost
Server        : SQLserver 2000
Description   : Post data as coming from an HTML FORM with METHOD=POST
                to an URL and retrieve the result.
Parameters    : @URL : the url to use ( like https://www.d-trix.com )
                @post : the parameters to post
                @WebLogin : (optional) The Username for the webserver
                @WebPassword : (optional) The password for the webserver
                @ProxyLogin : (optional) The Username for the proxyserver
                @ProxyPassword : (optional) The password for the proxyserver
Notes         : . The data to be posted should be like ?param1=val1&par2=val2
    *****/

```

```

        . if a table called #usp_httppost exists, the result is stored
        into that table. If not, the procedure returns a resultset.
Date       : 2005-01-19
Author     : Bert De Haes ( bertdehaes@scarlet.be )
History    :
*****/

set nocount on

declare @http int, -- the objecttoken for WinHttp.WinHttpRequest.5.1
        @rc int, -- the return code from sp_OA procedures
        @src varchar(255), -- the source of an error
        @desc varchar(512), -- the description of an error
        @Doing varchar(512), -- What are we doing when calling a sp_OA proc
        @TableExisted bit -- Did the temp table exists yes(1) or no(0)

-- init

set @rc = 0

if object_id('tempdb..#usp_httppost') is null
begin
    -- The temp table #usp_httppost does not exists
    set @TableExisted = 0
    create table #usp_httppost
    (
        return_status bit null, -- 0 is OK, 1 is error
        error_msg varchar(4096) null,
        HTTP_Status varchar(30) null,
        HTTP_AllResponseHeaders text null,
        HTTP_ResponseText text null
    )
end
else
begin
    set @TableExisted = 1
    truncate table #usp_httppost
end

-- Insert a default record

insert #usp_httppost ( return_status, error_msg )
values ( 1 , 'Unknown error' )

-- create a table to store output from different sp_OA calls

create table #tempresult
(
    HTTP_Status varchar(30) null,
    HTTP_AllResponseHeaders text null,
    HTTP_ResponseText text null
)

```

```

-- create the 'WinHttp.WinHttpRequest.5.1' object

set      @Doing = 'Create WinHttp.WinHttpRequest.5.1 object.'
exec      @rc      = master.dbo.sp_OACreate 'WinHttp.WinHttpRequest.5.1', @http
output
if @rc <> 0 goto Error

-- open the url on the server

set      @Doing = 'Open("POST" , "" + @URL + "", 0 )'
exec      @rc      = master.dbo.sp_OAMethod @http, @Doing
if @rc <> 0 goto Error

-- set the SetRequestHeader

set      @Doing = 'SetRequestHeader("Content-Type","application/x-www-form-
urlencoded")'
exec      @rc      = master.dbo.sp_OAMethod @http,@Doing
if @rc <> 0 goto Error

if @WebLogin is not null AND @WebPassword is not null
begin
    -- Set the Credentials for the Webserver

    set      @Doing = 'SetCredentials("'" + @WebLogin + "','" +
@WebPassword + "','0)')
    exec      @rc      = master.dbo.sp_OAMethod @http, @Doing
    if @rc <> 0 goto Error
end

if @ProxyLogin is not null AND @ProxyPassword is not null
begin
    -- Set the Credentials for the Proxy

    set      @Doing = 'SetCredentials("'" + @ProxyLogin + "','" +
@ProxyPassword + "','1)')
    exec      @rc      = master.dbo.sp_OAMethod @http, @Doing
    if @rc <> 0 goto Error
end

-- send the info

set      @Doing = 'Send("'" + @post + "'" )'
exec      @rc      = master.dbo.sp_OAMethod @http,@Doing
if @rc <> 0 goto Error

-- Get the HTTP_Status

set      @Doing = 'Status'
truncate table #tempresult

```

```

insert  #tempresult (HTTP_Status)
exec    @rc = sp_OAMethod @http,@Doing
if @rc <> 0 goto Error

update  #usp_httppost
set     HTTP_Status = #tempresult.HTTP_Status
from    #tempresult
where   #tempresult.HTTP_Status is not null

-- Get the ResponseHeaders

set     @Doing = 'GetAllResponseHeaders'
truncate table  #tempresult
insert  #tempresult (HTTP_AllResponseHeaders)
exec    @rc = sp_OAMethod @http,@Doing
if @rc <> 0 goto Error

update  #usp_httppost
set     HTTP_AllResponseHeaders = #tempresult.HTTP_AllResponseHeaders
from    #tempresult
where   #tempresult.HTTP_AllResponseHeaders is not null

-- retrieve the ResponseText

set     @Doing = 'ResponseText'
truncate table  #tempresult
insert  #tempresult (HTTP_ResponseText)
exec    @rc = sp_OAMethod @http,@Doing
if @rc <> 0 goto Error

update  #usp_httppost
set     HTTP_ResponseText = #tempresult.HTTP_ResponseText
from    #tempresult
where   #tempresult.HTTP_ResponseText is not null

-- Everything went well

update #usp_httppost
set return_status = 0,error_msg = 'DONE'

-- if we get here the normal way, don't do error
GOTO Cleanup

Error:

-- Get error information

if @http is not null
begin
    exec sp_OAGetErrorInfo @http, @src OUT, @desc OUT
end
else

```

```

begin
    set @src = '?'
    set @desc = '?'
end

update #usp_httppost
set     return_status = 1,
        error_msg =
            'Error [' + ISNULL( master.dbo.fn_hexadecimal
(@rc)      , '' ) +
            '], While [' + ISNULL
( @Doing      , '' ) +
            '], Source [' + ISNULL
( @src      , '' ) +
            '], Description [' + ISNULL
( @desc      , '' ) + ']'

-- Destroy created object(s)

Cleanup:
if @http is not null
begin
    exec      @rc      = master.dbo.sp_OADestroy @http
    set @http = null
    if @rc <> 0 goto Error
end

-- Give the result back to the caller

Result:
if @TableExisted = 0
    select * from #usp_httppost

return 0
end

```