

Caracas, 14 de Agosto de 2007

Mejora en el almacenamiento de los correos electrónicos dentro de N2C

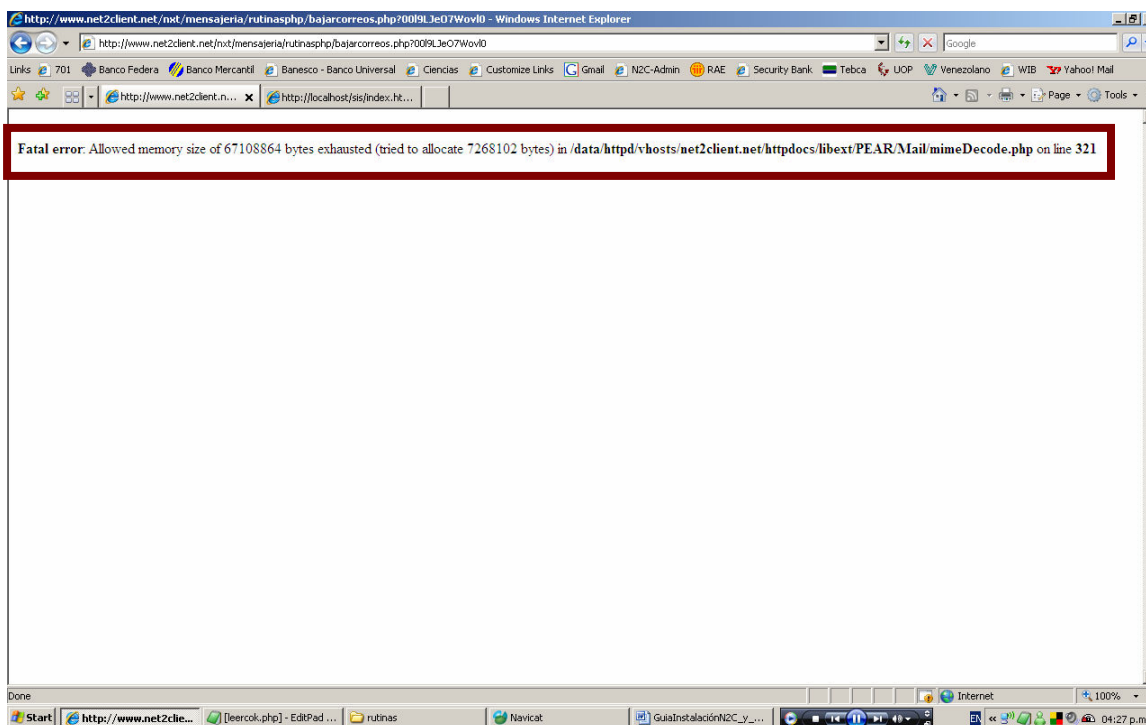
Mejoras hechas a los programas:

- *nxt/mensajeria/rutinasphp/bajarcorreos.php*

En este programa, en la línea 33 se agregó la instrucción:

`ini_set("memory_limit","256M");`

Esto se hizo porque el programa consumía mucha memoria en tiempo de ejecución al momento de abrir los correos, arrojando un **Fatal Error** de php, como muestra la siguiente imagen:



El error surge en la línea **321** del programa ***mimeDecode.php***, ya que es en esta línea donde el programa carga en memoria los archivos adjuntos de un correo.

Adicionalmente, en el programa ***nxt/mensajeria/rutinasphp/fun_email.php***, específicamente en la función GuardarAdjuntos; se utilizaba la función *bin2hex()* de php, para codificar el contenido de los archivos adjuntos y así crear un Query que fuese seguro para ser insertado en mysql. Dicha función -bin2hex()- generaba cadenas de datos de más del doble del tamaño del archivo adjunto original que se estuviese codificando en ese momento. De forma tal, que si se recibía un email con un attachment de 10 megas, el programa alojaba una cadena de 20 megas o más, para poder ser almacenada en MySql. Esto hacía que el programa fuese sumamente ineficiente en cuanto a manejo de memoria, por lo que se realizaron los ajustes que se describen a continuación.

- *nxt/mensajeria/rutinasphp/func_email.php*

En este programa se modificó casi por completo la función *GuardarAdjunto(\$NombreOriginal,\$ContAttachments)*, en donde, en primer lugar, se quitó la forma como se codificaban los adjuntos (*bin2hex(\$ContAttachments)*) y simplemente se sustituyó por la función *addslashes(\$ContAttachments)*, la cual no genera cadenas de datos innecesariamente grandes y funciona eficientemente para guardar distintos tipos de datos.

Como consecuencia, el programa cesó de arrojar errores de php. No obstante, al ejecutar el Query que guarda un attachment a la tabla EEEXMA, MySql indicaba el siguiente mensaje de error: *se recibió una cantidad mayor de paquetes a los permitidos* (traducción aproximada del inglés).

Como solución a este nueva situación, se escribió un algoritmo que segmentara un archivo adjunto cualquier, mayor de 750KB (valor arbitrario, menor de 1MB), y guardara dicho archivo adjunto por bloques, realizando primero un INSERT con los primeros 750KB y luego tantos UPDATE's como fuesen necesarios, de 750KB cada uno, para almacenar el resto de los datos sobre el registro recién insertado (líneas **132** hasta **152** del programa *nxt/mensajeria/rutinasphp/func_email.php*).

La forma en que se anexa datos a una columna con dichos UPDATE's, es utilizando la función CONCAT(Columna,"Nuevo datos"), la cual es provista por MySql y se describe a continuación:

```
$x="UPDATE EEEXMA SET  
XMAContenido=CONCAT(XMAContenido,\"\".addslashes(substr($ContAttachments,$offset,$TamBloque)).\"")  
WHERE XMACodigo=$XMACodigo";  
//Donde: $ContAttachment es el contenido completo del attachment, $offset es el comienzo del bloque  
// y $TamBlouqe el tamaño arbitrario definido para un bloque, en este caso 750KB
```

De forma tal que se concatenan los datos alojados en una celda específica, en este caso XMAContenido con los que se indiquen en el Query (2do parámetro de la función CONCAT()).

A pesar de ser esta una solución lógicamente correcta, MySql presenta una falla interna en la implementación de la función CONCAT, la cual se encuentra descrita en detalle en el enlace a continuación:

<http://bugs.mysql.com/bug.php?id=20458>

Como se puede concluir al leer la página anteriormente referida, la única solución propuesta hasta el momento para solucionar el error de implementación del CONCAT en MySql, es aumentar el valor del parámetro *max_allowed_packets* en tiempo de ejecución, el cual indica

el tamaño máximo que puede tener cualquier Query que se envíe a MySQL, y cuyo valor por defecto es 1MB. Se supone que esta variable fue modificada en el archivo de configuración de MySQL en el servidor www.net2client.net; pero por algún motivo este cambio no tuvo efecto, como se pudo comprobar al averiguar el valor de esta variable en momento de ejecución (ver código a continuación):

```
$z="select @@max_allowed_packet";  
$y=mysql_query($z);  
$result=mysql_fetch_assoc($y);  
//print_r($result);// Devuelve 1MB en bytes  
//Cambia el valor de max_allowed_packet en tiempo de ejecución  
$r="set session max_allowed_packet=1024*1024*50;";  
$p=mysql_query($r);  
$result=mysql_fetch_assoc($p);  
//print_r($result);//Devuelve el nuevo valor de 50MB en bytes
```

De forma tal que al aumentar el tamaño de los paquetes permitidos por MySQL, el error no se arroja más y los datos son guardados sin problema a la tabla EEEXMA.

Es importante destacar que las soluciones propuestas en el presente documento no necesitan que se editen los archivos de configuración de la plataforma (php.ini, httpdconfig, etc) sino que se soluciona por código fuente de los siguientes programas únicamente:

1. ***nxt/mensajeria/rutinasphp/bajarcorreos.php***
2. ***nxt/mensajeria/rutinasphp/func_email.php***



Enrique A. Areyán Viqueira

Director

www.azquality.net