



Blog Cosillas de Juansal

El blog en ebook

En este libro iré introduciendo todas las entradas que he realizado en mi blog personal.

Juan Salvador Aleixandre Talens



Blog Cosillas de Juansal

Juan Salvador Aleixandre Talens

Este libro está a la venta en <http://leanpub.com/blogcosillasdejuansal>

Esta versión se publicó en 2018-01-09



Este es un libro de [Leanpub](#). Leanpub anima a los autores y publicadoras con el proceso de publicación. [Lean Publishing](#) es el acto de publicar un libro en progreso usando herramientas sencillas y muchas iteraciones para obtener feedback del lector hasta conseguir tener el libro adecuado.

© 2015 - 2018 Juan Salvador Aleixandre Talens

Índice general

Blog Cosillas de Juansal	i
2010	1
Presentación	1
Migración de RX Components a JVCL I	1
Migración de RX Components a JVCL 0	2
Activar la cuenta root en Ubuntu	2
Reseteo del ID y de la MAC de una máquina virtual en VMware	2
Hacer que el VMware utilice la memoria física y no un fichero en una máquina virtual	3
Crear un mirror de un repositorio de subversion	3
Lectura de ficheros INI sin secciones con Inno Setup desde código	4
Lectura de ficheros INI sin secciones con Inno Setup desde código (continuación)	6
Desactivación de Unity de las máquinas virtuales de VMware Workstation	7
2011	8
Plugin de Firefox	8
Cambiar el propietario y el grupo al que pertenece un fichero o una carpeta en linux	8
Obtener información de la máquina y del Windows instalado en ella desde línea de comandos	9

Blog Cosillas de Juansal

En este libro iré introduciendo todas las entradas que he realizado en mi [blog personal](http://blog.juansal.com)¹.

Las entradas del blog estarán agrupadas por años y ordenadas por fecha de introducción en el blog.

Además de las entradas del blog introduciré el resto de páginas del blog en un capítulo llamado secciones adicionales.

¹<http://blog.juansal.com>

2010

En esta página he creado una relación de las entradas del blog con su fecha de publicación.

Fecha	Entrada
21 de abril de 2010	Presentación
27 de abril de 2010	Migración de RX Components a JVCL I
28 de abril de 2010	Migración de RX Components a JVCL 0
4 de mayo de 2010	Activar la cuenta root en Ubuntu
19 de mayo de 2010	Reseteo del ID y de la MAC de una máquina virtual en VMware
19 de mayo de 2010	Hacer que el VMware utilice la memoria física y no un fichero en una máquina virtual
29 de julio de 2010	Crear un mirror de un repositorio de subversion
15 de septiembre de 2010	Lectura de ficheros INI sin secciones con Inno Setup desde código
16 de septiembre de 2010	Lectura de ficheros INI sin secciones con Inno Setup desde código (continuación)
21 de octubre de 2010	Desactivación de Unity de las máquinas virtuales de VMware Workstation

Presentación

La finalidad de este blog es introducir lo que me parezca, en un principio no va a ser dirigido a ningún tema en especial.

Migración de RX Components a JVCL I

En estas entradas voy a poner los componentes del RX y su equivalente en JVCL. En este primer artículo sólo voy a poner el *TFormPlacement*. El equivalente a este en JVCL es *TJvFormStorage*.

Los métodos de ambos son los mismos. El segundo es igual que el primer con más parámetros y métodos, pero los métodos y parámetros del primero están en el segundo. La única novedad que hay que señalar es que para que guarde necesitamos un *AppStorage*, los que hay disponibles son:

- TJvAppIniFileStorage
- TJvAppRegistryStorage
- TJvAppDBStorage

- `TJvAppXMLFileStorage`

Con sólo asignar uno de estos *AppStorage* al *TJvFormStorage* y configurarlo funciona.

Migración de RX Components a JVCL 0

Este post debería de haber aparecido antes del actual, pero por despiste no lo hice y por tanto lo he etiquetado como 0 y no como II. Para realizar la migración, esta se puede hacer de forma automática. Esto se realiza mediante una aplicación que tiene JVCL llamada *JVCLConvert*, esta se encuentra en la ruta: `javcl\devtools\JVCLConvert`.

Para poder realizar la conversión de componentes de forma correcta hay que utilizar plantillas, estas se encuentran en `javcl\converter` y las que nos interesan son las siguientes:

- `RxToJVCLApp.dat`
- `JVCL3.dat`

La primera sirve para realizar la conversión de nombres de clases y la segunda sirve para pasar los nombres de clases de cualquier versión anterior de JVCL a la actual. Hay que indicar que aunque la segunda no es obligatoria, sí que es aconsejable, ya que la plantilla primera podría tener nombres de clases antiguas de JVCL y por tanto la segunda se encarga de actualizar estos nombres.

Otra cosa que no indica la documentación y que sí que es recomendable es que una vez transformado el proyecto lo abramos y abramos todos sus formularios ya que puede darse el caso, y a mí se me ha dado, que algún **USES** se haya quedado sin incluir, ya que esta aplicación cambia nombres, pero no añade y puede que alguna clase no tenga incluido su **USES**.

Activar la cuenta root en Ubuntu

Para activar la cuenta root de Ubuntu se ha de lanzar el siguiente comando. `bash sudo -u root passwd` y pones el pass del usuario que estas utilizando, después de eso te pedirá el nuevo pass para el root. De esta forma ya está activado el usuario. Después de esto ya se puede lanzar el comando `bash su -` para entrar a nuestra cuenta de root.

Reseteo del ID y de la MAC de una máquina virtual en VMware

Para resetear el ID de una máquina virtual se han de borrar las líneas que contengan las siguientes entradas:

```
1 uuid.location
2 uuid.bios
```

Para resetear la MAC de la tarjeta de red hay que borrar las siguientes líneas

```
1 ethernet0.generatedAddress
2 ethernet0.generatedAddressOffset
```

Además es obligado que esta línea tenga este valor `ethernet0.addressType = "generated"`.

Hacer que el VMware utilice la memoria física y no un fichero en una máquina virtual

Para conseguir esto hay que modificar el fichero .vmx de la máquina virtual y añadir/modificar las siguientes entradas:

```
1 mainMem.useNamedFile = "FALSE"
2 MemAllowAutoScaleDown = "FALSE"
3 MemTrimRate = "0"
4 sched.mem.pshare.enable = "FALSE"
```

Crear un mirror de un repositorio de subversion

Para crear un mirror de un repositorio de subversion se siguen los siguientes pasos. La mayoría de los pasos se hacen sobre el mirror.

Creamos el repositorio en la máquina del mirror. Creamos un usuario en ambos servidores para realizar la sincronización (en este ejemplo el usuario creado ha sido: syncuser) Lanzamos los siguientes comandos en el mirror:

```
1 svnsync initialize file:///ruta_local_mirror ruta_servidor_svn --username syncu\
2 ser --password syncuser
3 svn proplist --revprop -r 0 file:///ruta_local_mirror
4 svn propget svn:sync-from-url --revprop -r 0 file:///ruta_local_mirror
5 svn propdel svn:sync-lock --revprop -r 0 file:///ruta_local_mirror
```

Con esto ya tenemos el mirror montado y preparado para funcionar. Un comando interesante a ejecutar, aunque no necesario, es el siguiente, el cual se encarga de la sincronización inicial de los repositorios. Como he dicho no es necesario, pero si el repositorio del que vamos a hacer el mirror

ya contiene información es recomendable, especialmente si este ya tiene muchos commits realizados. `bash svnsync --non-interactive sync file:///ruta_local_mirror --username syncuser --password syncuser` Después de esta sincronización inicial es conveniente ejecutar el siguiente comando para que se copien las propiedades del repositorio (sólo es necesario esta primera vez): `bash svnsync copy-revprops file:///ruta_local_mirror --username syncuser --password syncuser`

Después de esto en el repositorio origen del subversion se puede crear un hook para que la sincronización se realice de forma automática, el hook en el que incluir el siguiente comando es el `post_commit`: `bash svnsync --non-interactive sync ruta_trabajo_mirror --username syncuser --password syncuser`

Lectura de ficheros INI sin secciones con Inno Setup desde código

En este post voy a poner un código de ejemplo en el cual se parsea un fichero INI sin secciones (fichero INI estilo UNIX) desde el Inno Setup. Para esto hay que crear el código en la sección [Code] de este programa.

```
1  [Code]
2  var
3      ConfigValues:    TArrayOfString;
4
5  function LeeFicheroConfiguracion (FileName: String): Boolean;
6  begin
7      Result := LoadStringsFromFile (FileName, ConfigValues);
8  end;
9
10 function ObtieneValorEtiqueta (Etiqueta: String; var Value: String): Boolean;
11 var
12     I:      LongInt;
13     L:      LongInt;
14     S:      String;
15 begin
16     S := Etiqueta + '=';
17     L := Length (S);
18     For I := 0 to GetArrayLength (ConfigValues) - 1 do
19         begin
20             if (copy (ConfigValues [I], 1, L) = S) then
21                 begin
22                     Value := copy (ConfigValues [I], L + 1,
```



```
23         Length (ConfigValues [I]) - L);
24         Result := TRUE;
25         Exit;
26     end;
27 end;
28 Result := FALSE;
29 end;
30
31 function EscribeValorEtiqueta (Etiqueta: String; Value: String): Boolean;
32 var
33     I:      LongInt;
34     L:      LongInt;
35     A:      LongInt;
36     S:      String;
37 begin
38     S := Etiqueta + '=';
39     L := Length (S);
40     A := GetArrayLength (ConfigValues);
41     For I := 0 to A - 1 do
42     begin
43         if (copy (ConfigValues [I], 1, L) = S) then
44         begin
45             ConfigValues [I] := S + Value;
46             Result := TRUE;
47             Exit;
48         end;
49     end;
50     SetArrayLength (ConfigValues, A + 1);
51     ConfigValues [A] := S + Value;
52     Result := FALSE;
53 end;
54
55 function EscribeFicheroConfig (FileName: String): Boolean;
56 var
57     bRet:    Boolean;
58 begin
59     // FileName -> Backup.
60     bRet := FileCopy (FileName,
61         ExpandConstant ('{tmp}\' + ExtractFileName (FileName)),
62         FALSE);
63     if (bRet) then
64     begin
```

```
65         bRet := SaveStringsToFile (FileName, ConfigValues, FALSE);
66         if (bRet) then
67         begin
68             Result := TRUE;
69             Exit;
70         end else
71         begin
72             // Maybe the backup file should be copied back here?
73         end;
74     end;
75     Result := FALSE;
76 end;
```

Espero que le sirva a la gente que haya tenido el mismo problema que yo.

Lectura de ficheros INI sin secciones con Inno Setup desde código (continuación)

Este post es una continuación del anterior y voy a añadir un nuevo método, el cual se encarga de comprobar si una etiqueta existe en el fichero.

```
1  function ExisteEtiqueta (Etiqueta: String): Boolean;
2  var
3      I:      LongInt;
4      L:      LongInt;
5      S:      String;
6  begin
7      S := Etiqueta + '=';
8      L := Length (S);
9      For I := 0 to GetArrayLength (ConfigValues) - 1 do
10     begin
11         if (copy (ConfigValues [I], 1, L) = S) then
12         begin
13             Result := TRUE;
14             Exit;
15         end;
16     end;
17     Result := FALSE;
18 end;
```

Desactivación de Unity de las máquinas virtuales de VMware Workstation

Al desactivar el unity de las máquinas virtuales conseguimos que la carpeta cache que ha aparecido desde la versión 7.1 desaparezca.

En el fichero config.ini hay que introducir/modificar la siguiente línea:

```
1 isolation.tools.unity.disable = "true"
```

En el fichero vmx de la máquina virtual:

```
1 unity.enableLaunchMenu = "FALSE"  
2 unity.showBadges = "FALSE"  
3 unity.showBorders = "FALSE"  
4 unity.wasCapable = "FALSE"
```

2011

En esta página he creado una relación de las entradas del blog con su fecha de publicación.

Fecha	Entrada
24 de marzo de 2011	Plugin de Firefox
5 de julio de 2011	Cambiar el propietario y el grupo al que pertenece un fichero o una carpeta en linux
8 de noviembre de 2011	Obtener información de la máquina y del Windows instalado en ella desde línea de comandos

Plugin de Firefox

Ya he creado mi primer plugin de Firefox. Ese es muy simple, ya que, lo que hace es mostrar un botón en la barra de complementos con el icono de Mozilla (provisionalmente, hasta que encuentre uno que me guste) el cual abre el blog en una pestaña nueva.

Se que esto no es nada del otro mundo, pero por algo se empieza.

Cuando lo suba a algún sitio lo pondré en [la página del blog que he creado para el/los plugin/s²](#).

Cambiar el propietario y el grupo al que pertenece un fichero o una carpeta en linux

Propietario

Para cambiar el propietario de un fichero se utiliza el siguiente comando: `bash chown Nuevo_Propietario fichero`

Para cambiar el propietario de una carpeta: `bash chown Nuevo_Propietario carpeta`

Para cambiar el propietario de una carpeta y su contenido de forma recursiva: `bash chown -R Nuevo_Propietario carpeta`

²<http://blog.juansal.com/p/plugins-de-firefox.html>

Grupo

Para cambiar el grupo de un fichero se utiliza el siguiente comando: `bash chgrp Nuevo_Grupo fichero`

Para cambiar el grupo de una carpeta: `bash chgrp Nuevo_Grupo carpeta`

Para cambiar el grupo de una carpeta y su contenido de forma recursiva: `bash chgrp -R Nuevo_Grupo carpeta`

Propietario y grupo

Para cambiar el propietario y el grupo de un fichero se utiliza el siguiente comando: `bash chown Nuevo_Propietario:Nuevo_Grupo fichero`

Para cambiar el propietario y el grupo de una carpeta: `bash chown Nuevo_Propietario:Nuevo_Grupo carpeta`

Para cambiar el propietario y el grupo de una carpeta y su contenido de forma recursiva: `bash chown -R Nuevo_Propietario:Nuevo_Grupo carpeta`

Obtener información de la máquina y del Windows instalado en ella desde línea de comandos

Para obtener la información de una máquina (ordenador) y del windows instalado en ella existe un comando desde la línea de comandos llamado: `bash systeminfo.exe` Este comando devuelve, entre otros:

1. Nombre del host e información del usuario
2. Nombre del sistema operativo
3. Versión del sistema operativo
4. Información del sistema operativo y de las revisiones instaladas
5. Información de la placa base
6. Información de memoria
7. Información de discos
8. Información de red