

Documentación Scripts exportación histogramas Philips Pinnacle3

A. Pérez Rozos

alberto.perez@uma.es / alberto.perez.sspa@juntadeandalucia.es

Noviembre 2017

El objetivo de estas notas, un tanto informales, es explicar los scripts realizados en el planificador Philips Pinnacle para la exportación (y posterior importación) de histogramas dosis volumen con el fin de poder ser utilizados y modificados por otros centros.

Notas previas para los scripts de HDV y de datos de planificación si se desean instalar en un servidor que no sea el centralizado o en instituciones fuera del SAS:

Los scripts se instalarán en la carpeta de scripts de pinnacle /usr/local/PinnacleSiteData/clinical/Scripts/ dentro de una subcarpeta MAVV.

En el directorio de usuarios /home/ debemos crear una carpeta pinmalaga/PRESCRIPCIONES/, dentro de esta última carpeta es donde se realizarán las exportaciones y de donde se recuperarán los datos exportados.

Los scripts de unix deben tener privilegios de ejecución, y los directorios PRESCRIPCIONES, MAVV y sus subdirectorios privilegios de lectura y de escritura.

El conjunto de scripts se compone de los siguientes tres scripts:

- DVHAusgabe.Script.bj (/MAVV/
- BuildDVHScript (/MAVV/tmpDVH/)
- DVHExport.script.auto (/MAVV/)

Siendo el último de ellos generado en cada ocasión por los otros dos y personalizado para cada paciente, por lo que no es estrictamente necesario y únicamente se ha incluido en el listado por claridad.

Estructura de directorios:

El directorio de scripts de trabajo para MAVV es

/usr/local/PinnacleSiteData/clinical/Scripts/MAVV

Dentro de ese directorio debe haber creado otro directorio tmpDVH donde se ubica el script BuildDVHScript, y a su vez otro directorio tmpDVH/tmp donde se realizan cálculos adicionales temporales.

El Script se inicia con DVHAusgabe.Script.bj , que es una adaptación de un script proporcionado por Bjorne Riis.

Este script se encuentra ubicado en el directorio /usr/local/PinnacleSiteData/clinical/Scripts/MAVV, pero podría ubicarse en el directorio general de scripts con sólo editar los paths correspondientes. (MAVV corresponde con el código establecido para el Hospital Virgen de la Victoria).

```

C:\Users\hbrad03\Desktop\2017Basura\PinnCentralizado\MAVV\DVHAusgabe.Script.bj - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
[README] [ana4do3d] [PenEasyLinPlus] [squash] [ImportaCubeDoseB.Script] [desktop] [DVHAusgabe.Script.bj]
1 //WaitMessage = " writing Prescription";
2 //Script.ExecuteNow = "/usr/local/adacnew/PinnacleSiteData/Scripts/PrescriptionExport.Script.k1" ;
3 //WaitMessageOff = "";
4
5 WaitMessage = " writing DVH";
6 DVHList.Save = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/DVH.txt";
7 WaitMessageOff = "";
8
9
10 WaitMessage = " building DVH Data Script";
11 SpawnCommand = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/buildDVHScript";
12 WaitMessageOff = "";
13
14 WaitMessage = " writing DVH Data";
15 Script.ExecuteNow = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/DVHExport.Script.auto" ;
16 WaitMessageOff = "";
17
18 //ahora copiamos los archivos al directorio PRESCRIPCIONES
19
20 Store.At.Command = SimpleString();
21 Store.At.Command2 = SimpleString();
22 Store.At.Command2.String = "mkdir /home/pinnalaga/PRESCRIPCIONES/";
23 Store.At.Command2.AppendString = PlanInfo.MedicalRecordNumber;
24 SpawnCommand = Store.At.Command2.String;
25 //WarningMessage = Store.At.Command2.String;
26 Store.At.Command.String = "cp /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/* /home/pinnalaga/PRESCRIPCIONES/";
27 Store.At.Command.AppendString = PlanInfo.MedicalRecordNumber;
28 Store.At.Command.AppendString = "/";
29 SpawnCommand = Store.At.Command.String;
30 WarningMessage = Store.At.Command.String;
31
32 //WaitMessage = " open OpenOffice";
33 //SpawnCommand = "cp /home/p3rtp/Export/DVH.gxg /home/p3rtp/Export/DVH.gxg";
34 //SpawnCommandNoWait = "/home/p3rtp/OpenOffice/soffice /home/p3rtp/Export/DVH.gxg";
35 //WaitMessageOff = "";

```

En la línea 6 se guarda la lista de volúmenes activados en la ventana de HDVs, una vez guardada esa lista, se invoca (línea 10) el script de UNIX buildDVHScript, situado en la carpeta tmpDVH

```

C:\Users\hbrad03\Desktop\2017Basura\PinnCentralizado\MAVV\tmpDVH\buildDVHScript - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
[README] [ana4do3d] [PenEasyLinPlus] [squash] [ImportaCubeDoseB.Script] [desktop] [DVHAusgabe.Script] [buildDVHScript]
1 #!/bin/csh
2 # DVH Ausgabe
3 # Bj 03.2005
4
5 set outScript = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/DVHExport.Script.auto"
6 set Import = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/DVH.txt"
7 set Pfad = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp"
8
9 # delete old Data
10 if (-e $Pfad/DVH.Data.0.txt ) rm $Pfad/DVH.Data*
11 if (-e $Pfad/DVH.Data.0.txt ) rm $Pfad/DVH*
12 if (-e $outScript ) rm $outScript
13
14 # List of ROI Names
15 # cat $Import | grep RegionOfInterestName | cut -f2 -d\" > /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/DVH.Name.csv
16 cat $Import | grep RegionOfInterestName | cut -f2 -d\" > /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/DVH.Name.txt
17
18 # Liste der Trial Namen
19 # cat $Import | grep TrialName | cut -f2 -d\" > /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/Trial.Name.csv
20 cat $Import | grep TrialName | cut -f2 -d\" > /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/Trial.Name.txt
21
22
23
24 #find Number of ROI
25 # Anzahl = `grep -c RegionOfInterestName $Import`
26
27
28 # write outputscript
29
30 echo "DVHPlotStyle .CumulativeY = 1 ;" > $outScript
31 echo "DVHPlotStyle .NormalizeX = 0;" > $outScript
32 echo "DVHList .Current .AutoComputeMaxDose = 1;" > $outScript
33
34 while ($Anzahl > 0)
35
36     @ Anzahl--
37     echo " DVHList.Current = $Anzahl ;" >> $outScript
38     echo " DVHList .Current .NumberOfBins = \"200\" ;" >> $outScript
39     # echo ' DVHList.Current.Data.Save = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/DVH.Data.'$Anzahl'.gxg\" ;' >> $outScript
40     echo ' DVHList.Current.Data.Save = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpDVH/tmp/DVH.Data.'$Anzahl'.txt\" ;' >> $outScript
41 end
42
43 rm $Import
44
45
46

```

En las primeras líneas de este script se definen las variables outScript, Import y Pfad que definen los directorios de trabajo temporales dónde operará el resto del script. Es importante, para no cometer errores, que estas carpetas no sean comunes a otros procedimientos que hagamos en Pinnacle, ya que ahí borraremos archivos temporales (líneas 9-12) y siempre queda la posibilidad de que borrásemos otro tipo de archivos.

[illegible]

En este punto en el directorio ~/PRESCRIPCIONES estarán guardados los archivos que contienen los nombres de cada una de las regiones, los trials al que pertenecen, y tantos archivos DVH.Data.n.txt como regiones hayamos exportado.

Una vez ahí se ejecuta la hoja Excel de importación de histogramas y planificaciones. Actualmente esta hoja realiza la importación por ftp, pero el servidor de pinnacle en su configuración centralizada actual sólo permite el acceso por sftp, habría que reconfigurar Excel (y no se si lo permite) para acceder por sftp a dicha carpeta. Si esto no fuera posible habría que primero enviar esos archivos a alguna carpeta local y posteriormente importarlos con la hoja/aplicación...

Documentación Scripts exportación datos técnicos de la planificación Philips Pinnacle3

A. Pérez Rozos
Noviembre 2017

El objetivo de estas notas, un tanto informales, es explicar los scripts realizados en el planificador Philips Pinnacle para la exportación (y posterior importación) de los datos técnicos de una planificación para, por ejemplo, generar una ficha de tratamiento y realizar un cálculo independiente de unidades de monitor, con el fin de poder ser utilizados y modificados por otros centros.

El conjunto de scripts se compone de los siguientes scripts:

- ExportarIMRT.Script.p3rtp (/MAVV/)
- ExportaHacesInicio.Script (/MAVV/)
- ExportaHacesGeneral.Script (/MAVV/)
- ExportaHacesUnix (/MAVV/)
- ExportaSSD.Script (/MAVV/)
- buildSSD.Script (/MAVV/tmpSSD/)
- SSDExport.Script.auto (/MAVV/)
- buildSSD2.Script (/MAVV/tmpSSD/)

Estructura de directorios:

El directorio de scripts de trabajo para MAVV es

/usr/local/PinnacleSiteData/clinical/Scripts/MAVV

Dentro de ese directorio debe haber creados otros directorios tmpRV y tmpSSD.

Trabajan en el directorio de Scripts de la institución MAVV, aunque esto puede cambiarse sin más que cambiar las referencias a los paths en los sitios correspondientes.

```
1 // APerezRozos
2 // 2013 Exportacion de planes IMRT y arcsos
3 //Primero rellenos los fieldID de los campos
4 //Script.ExecuteNow = "/usr/local/adacore/PinnacleSiteData/Scripts/quitarar.Script.p3rtp";
5 //WindowList .TrialMain .Create = "PlanButton";
6 //WindowList .Export .Create = "Export...";
7 //Exportamos los datos de los campos de IMRT o generales
8 // (Migrado a PinnCentralizado 2017.09.25)
9 Script.ExecuteNow = "/usr/local/adacore/PinnacleSiteData/Scripts/MAVV/ExportaHacesInicio.Script";
10 //Guarda los puntos
11 Store.At.Command = SimpleString();
12 Store.At.Command.String = "/home/pinnacle/PRESCRIPCIONES/p";
13 Store.At.Command.AppendString = PlanInfo.MedicalRecordNumber;
14 Store.At.Command.AppendString = ".xxx";
15 Poilist .Save = Store.At.Command.String;
16 // Exporta las distancias de colocacion a cero grados
17 Script.ExecuteNow = "/usr/local/adacore/PinnacleSiteData/Scripts/MAVV/ExportarSSD.Script";
18 // Guarda todos los datos del TRIAL
19 Store.At.Command = SimpleString();
20 Store.At.Command.String = "/home/pinnacle/PRESCRIPCIONES/t";
21 Store.At.Command.AppendString = PlanInfo.MedicalRecordNumber;
22 Store.At.Command.AppendString = ".xxx";
23 TrialList.Current.Save = Store.At.Command.String;
24 // Ahora borra los archivos no necesarios
25 Store.At.MyBorrar=SimpleString();
26 Store.At.MyBorrar.String = "/home/pinnacle/PRESCRIPCIONES/t";
27 Store.At.MyBorrar.AppendString = PlanInfo.MedicalRecordNumber;
28 Store.At.MyBorrar.AppendString = ".xxx.binary*";
29 SpawnCommand = Store.At.MyBorrar.String;
30 // Quitamos los fieldID ya estan rellenos
31 // Script.ExecuteNow = "/usr/local/adacore/PinnacleSiteData/Scripts/quitarfieldID.Script.p3rtp";
32
```


El script `ExportaIMRT.Script.p3rtp` es el que inicia el proceso de exportación, se encarga de ir llamando al resto de Scripts y de ir moviendo la información a las carpetas de exportación.

En la línea 9 llama al Script `ExportaHacesInicio.Script` que es el encargado de iniciar el bucle de exportación de parámetros de los campos: (A su vez llamará a `ExportaHacesGeneral.Script` y a `ExportaHacesUnix`). Este script realiza cálculos temporales en `~/tmpRV` que debe estar creado desde el momento de instalación de los Scripts.

```
1 // -----
2 // Exportacion campos de tratamiento:
3 // * Leemos los datos generales del haz
4 // * Volcamos al fichero de salida los datos del haz
5 // * Reseteamos variables
6 // A.PerezRozos 2013 - MUVV
7 // -----
8 // Este Script se inicia con ExportaHacesInicio.Script
9 // luego llamara a ExportaHacesGeneral.Script
10 // y finaliza con el Script de UNIX ExportaHacesUnix
11 // -----
12 //
13 // Trabaja en el directorio Gammex
14 // Genera archivos temporales en ese directorio que posteriormente son borrados por el script
15 // (AADatosHaces.txt, AADatosUnHaz.txt, AAAdatos.txt)
16
17 WaitMessage = "EXPORTANDO DATOS DE CAMPOS - POR FAVOR ESPERE A QUE FINALICE EL SCRIPT";
18
19 TrialList.Current.BeginMessageBatch = "";
20
21 //Se crean las variables de almacenamiento
22 Store.At.MySaveFile = SimpleString();
23 Store.At.MySaveFile.String = "/home/pinnmalaga/PRESCRIPCIONES/dc";
24 Store.At.MySaveFile.AppendString = PlanInfo.MedicalRecordNumber;
25 Store.At.MySaveFile.AppendString = ".txt";
26
27 // Se borra la exportacion de cualquier paciente anterior
28 Store.At.BorraAnterior = SimpleString();
29 Store.At.BorraAnterior.String = "xm ";
30 Store.At.BorraAnterior.AppendString = Store.At.MySaveFile.String;
31 SpawnCommand= "xm /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosHaces.txt";
32 SpawnCommand= "xm /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosUnHaz.txt";
33 SpawnCommand= "xm /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AAAdatos.txt";
34 SpawnCommand= Store.At.BorraAnterior.String;
35
36 // Guardamos la identificacion del paciente
37 Store.At.MySaveFile.Save = "/usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosHaces.txt";
38 Store.StringAt.patname = PlanInfo.MedicalRecordNumber;
39 Store.At.patname.AppendString = " ";
40 Store.At.patname.AppendString = PlanInfo.LastName;
41 Store.At.patname.AppendString = " ";
42 Store.At.patname.AppendString = PlanInfo.FirstName;
43
44 // Incorporamos otros datos estadisticos
45 Store.At.patname.AppendString = " ";
46 Store.At.patname.AppendString = PlanInfo.Physician;
47 Store.At.patname.AppendString = " ";
48 Store.At.patname.AppendString = PlanInfo.Planner;
49 Store.At.patname.AppendString = " ";
50 Store.At.patname.AppendString = PlanInfo.PlanPath;
51 Store.At.patname.AppendString = " ";
52
53 Store.At.patname.Save = "/usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosUnHaz.txt";
54 SpawnCommand = "cat /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosUnHaz.txt >> /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosUnHaz.txt";
55
56 TrialList.Current.BeamList.ChildrenEachCurrent.#{@}.Script.ExecuteNow = "/usr/local/adaconex/PinnacleSiteData/Scripts/MAVV/ExportaHacesGeneral.Script";
57
58 //
59 // Invocamos un script para formatear el fichero de salida y posteriormente renombrar el fichero con un nombre decente
60 //
61 //
62 //
63 SpawnCommand= "/usr/local/adaconex/PinnacleSiteData/Scripts/MAVV/ExportaHacesUnix";
64
65 //
66 // el nombre de salida será dc+NHistoria.txt
67 // Es un archivo de datos separados por asteriscos (*)
68 //
69 //
70
71 Store.At.Renombrar = SimpleString();
72 Store.At.Renombrar.String = "cp /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AAAdatos.txt ";
73 Store.At.Renombrar.AppendString = Store.At.MySaveFile.String;
74 SpawnCommand= Store.At.Renombrar.String;
75
76 //Borramos ficheros inservibles
77 SpawnCommand= "xm /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosHaces.txt";
78 SpawnCommand= "xm /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosUnHaz.txt";
79 SpawnCommand= "xm /usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AAAdatos.txt";
80
81 TrialList.Current.EndMessageBatch = "";
82 WaitMessageOff = "";
83
84
```

En las líneas 21-28 se crean las variables para definir el lugar donde se almacenarán los datos. De la 27-34 se borran los datos de una posible exportación anterior.

Con la programación actual es posible que si se ejecuta a la vez el script en dos momentos del tiempo coincidentes (¿cuánto es coincidente en este ámbito?) se pueden mezclar las ejecuciones de los dos script generando algún tipo de problema. Hasta ahora, no se nos ha dado el caso en una sola institución con un número reducido de usuarios, en la nueva configuración no sabemos si podrá darse.

En las líneas 36-50 se va almacenando en una variable de pinnacle algunos datos identificativos y administrativos del paciente, que son guardados a un fichero de disco en las líneas 53-54. En estas dos últimas líneas se ha generado la cabecera del archivo que contendrá los datos de la planificación y que identifica al paciente, en la siguiente línea 57 se va a ir ejecutando en bucle el script `ExportaHacesGeneral.Script` en cada uno de los campos de la planificación. (Nota para mi: Seguramente tiene sentido incorporar el número de puntos de control de cada campo en la exportación, y posiblemente el MLC de cada punto de control(recuperar los scripts originales que se borraron en mi ausencia))

El script desde las líneas 15 a la 92 va almacenando en variables de Pinnacle la información que se consideró relevante para la generación de la ficha de tratamiento y del cálculo redundante de UM. En las líneas siguientes 98-166 se van concatenando los valores anteriores en una única variable de pinnacle (podría haberse hecho todo en un paso, pero posteriormente sería más difícil de debug), para por último guardar la variable en un archivo temporal `AADatosUnHaz.txt` que será volcado en el archivo que contiene los datos de todos los campos exportados anteriormente en el bucle `AADatosHaces.txt` (Estos dos archivos se encuentran, como ya se dijo, en el directorio `/tmpRV`). Al acabar su ejecución se devuelve el flujo al script anterior, que continuará con el bucle sobre los haces hasta pasar por todos ellos.

Una vez concluido el bucle sobre los haces el script `ExportaHacesInicio.Script` va a llamar en su línea 60 a un nuevo script para formatear el fichero final de salida y copiarlo al directorio de exportación. Este nuevo script será `ExportaHacesUnix`

```
1 #!/bin/csh
2 # Export beams
3 # AAAAAAAAAA 08.2013
4
5 set outScript = "/usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatos.txt"
6 set Import = "/usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpRV/AADatosHaces.txt"
7
8 cat $Import | grep String | cut -f2 -d\" >> $outScript
9
10
11 rm $Import
12
```

Es un script de unix que actuará sobre el fichero `AADatosHaces.txt` eliminando cadenas innecesarias generadas por pinnacle y guardando el texto importante en el archivo `AAAdatos.txt` (el lector a estas alturas se habrá dado cuenta de mi manía, sin mucho sentido, de poner varias ‘A’ delante de los nombres de algunos ficheros... lo siento), estas operaciones se hacen en la línea 8 con la combinación de comandos `cat/grep/cut`. Al acabar borra el fichero desde el que estaba leyendo (borra `AADatosHaces.txt`)

De vuelta a `ExportaHacesInicio.Script` renombramos y guardamos el fichero en su destino de exportación definitivo (líneas 67 a 74) y borramos los ficheros temporales (líneas 76 a 79). Se devuelve al flujo al script inicial `ExportaIMRT.Script` en su línea 10.

Los siguientes pasos van a ser la exportación de los puntos existentes en la planificación, las distancias foco-paciente a 0°, guardar todos los datos del trial (aunque creo que se dejaron de usar hace tiempo), y borrar ficheros innecesarios.

Vamos por orden: exportación de los puntos de la planificación y distancias SSD a 0°: En las líneas 10-15 se prepara el nombre del archivo que contendrá estos datos, se llama como el número de historia con una 'p' delante. En la línea 17 se llama al script `ExportarSSD.Script` que es el que inicia el bucle para la exportación.

Precaución: Este script está personalizado para tener un Clinac600C o al menos una máquina con esta geometría. En el estado actual no funcionará si no existe una máquina con ese nombre. habría que adaptarlo para trabajar con cualquier máquina, o bien sin ella (sólo faltaría la parte de distancias a 0° que solo se usa para automatizar las distancias de setup) (Nota a fecha 10/11/2017)

`ExportarSSD.Script`:

```
1 // Proposito: exportar la SSD para la colocacion del paciente
2 // Autor: APerezRozos
3 // Fecha: Julio 2007
4 //
5 // Abrimos la ventana de EvalDRR y si el paciente esta en prono
6 // Se ajusta la orientacion del crano a cero grados
7
8 TrialList.Current.PlanEvalDRRSetup = "Plan Eval DRRs...";
9 TrialList.Current.SetupBeamIsocenterName = "OTAC";
10 TrialList.Current.SetupBeamMachineName = "CLINAC600C";
11 TrialList.Current.PlanEvalDRR = "Plan Eval DRRs..";
12
13
14 WindowList.OrthoDRRs.Create = "Plan Eval DRRs...";
15 WaitMessage = "Espere, exportando DFSa0";
16 Store.At.Posicion=SimpleString();
17 Store.At.Posicion.String = "On front (prone)";
18 IF.PatientSetup.Position.STRINGEQUALTO.Store.At.Posicion.String.THEN.TrialList.##"0".QABeamList.SetupBeam2.StandardOrientation = "Posterior";
19 // BORRAMOS DATOS ANTIGUOS RESIDUALES
20 Store.At.BorrarSSDFile = SimpleString();
21 Store.At.BorrarSSDFile.String = "rm /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/SSDFile.txt";
22 SpawnCommand = Store.At.BorrarSSDFile.String;
23 Store.At.ConcatenarSSD = SimpleString();
24 Store.At.ConcatenarSSD.String = "cat /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/SSDtmp0.txt >> /usr/local/adacnew/PinnacleSiteData/Sc
25 // Ahora iniciamos el ciclo de exportacion de las distancias
26 //
27 PoiList.Save = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/PoiList.txt";
28 SpawnCommand = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/buildSSDScript";
29 Script.ExecuteNow = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/SSDExport.Script.auto";
30 SpawnCommand = "/usr/local/PinnacleSiteData/clinical/Scripts/MAVV/tmpSSD/buildSSD2Script";
31 // Ahora copiamos los datos al directorio de exportacion
32
33 Store.At.Command = SimpleString();
34 Store.At.Command.String = "cp /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/ListaSSD.txt /home/pinnalaga/PRESCRIPCIONES/SSD";
35 Store.At.Command.AppendString = PlanInfo.MedicalRecordNumber;
36 Store.At.Command.AppendString = ".txt";
37 SpawnCommand = Store.At.Command.String;
38 WaitMessageOff = "";
39 WindowList.OrthoDRRs.Destroy = "";
40
```

El script trabajará en el directorio temporal `MAVV/tmp/tmpSSD` que debe existir previamente (se creará al instalar los scripts).

El script comienza llamando a las ventanas de campos de simulación de pinnacle (líneas 8-11), en estas ventanas es posible extraer los datos de colocación sin tener que crear campos auxiliares ni modificar campos existentes.

Las distancias informadas en esta ventana se refieren a campos de setup nombrados según la orientación del paciente (por este y otros motivos es conveniente etiquetar a los pacientes en el TAC siempre head in supine... pero eso es otra batalla), lo primero es saber si el paciente está en prono o supino y esto lo hacemos con la condición IF/THEN de la línea 18. Si el paciente está en supino continuamos adelante, pero si está en prono seleccionamos un campo Posterior (respecto de anatomía de paciente, pero anterior respecto de geometría del acelerador) para el resto del proceso. Esta ventana exige que los campos estén asignados a una máquina. Por motivos históricos propios de la institución original se desarrolló para un clinac600C, pero podría cambiarse para seleccionar la máquina del primer campo o cualquier otra disponible (ya que no se selecciona por defecto).

En la líneas 20-22 se borran archivos temporales que pudieran quedar anteriormente, y en 23-24 se define la variable concatenarSSD que define el archivo /tmpSSD/SSDFile.txt y que será utilizada en el script que se generará automáticamente con el script unix siguiente.

en las líneas 27-30 se inicia el proceso de exportación de puntos y distancias. Primero (línea 27) guardamos la lista de puntos existentes en el planificador en el fichero PoiList.txt (en /tmpSSD), a continuación ejecutamos el script de UNIX buildSSDScript (situado en /tmpSSD)

```

1  #!/bin/csh
2  # Exportar las distancias SSD a cero grados
3  # APerezRozos Julio 2007
4
5  set outScript = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/SSDExport.Script.auto"
6  set Import = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/PoiList.txt"
7  set Pfad = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD"
8
9  # delete old Data
10 # if (-e $Pfad/PoiList.txt) rm $Pfad/PoiList.txt
11 if (-e $outScript) rm $outScript
12
13 #
14 #
15 #
16
17 #find Number of ROI
18 @ Anzahl = `grep -c 'Name' $Import`
19
20 #
21 #
22 #
23 #
24
25 while ($Anzahl > 0)
26
27     @ Anzahl--
28     #
29     echo 'Triallist. Current.QABeamList.#"1".Isocenter = PoiList.#"$Anzahl".Name;' >> $outScript
30     cat /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/plantillaSSD.txt >> $outScript
31 end
32
33
34 rm $Import
35

```

En línea 11 borramos el posible archivo existente anterior SSDExportScript.auto que será generado por este script particularizado para la lista de puntos actual. En la línea 18 se cuenta el número de puntos exportados (contenidos en PoiList.txt y referenciados por la variable Import) utilizando el comando grep. En las líneas 25-31 se

realiza un bucle en el que se va construyendo el script `SSDExport.Script.auto` particularizado para el número de puntos existente en la planificación.

Al acabar la ejecución de este scrip de Unix existirá el script `SSDExport.Script.auto` en la carpeta `/MAVV/`, y se continúa la ejecución en la línea 28 del script `ExportarSSD.Script` en la que se ejecuta `SSDExport.Script.auto`

Sólo para ilustrar el script:

```
1 TrialList.Current.QABeamList.#"1".Isocenter = PoiList.#"0".Name;
2 //
3 Store.StringAt.SSDtmp = SimpleString{};
4 Store.At.SSDtmp.String = TrialList.Current.QABeamList.SetupBeam2.SSD;
5 //WarningMessage = Store.At.SSDtmp.String;
6 Store.At.SSDtmp.Save = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/SSDtmp0.txt";
7 // Y ahora concatenamos el store con los datos anteriores
8 SpawnCommand =Store.At.ConcatenarSSD.String;
9 //
10 //
11
```

(mal ejemplo... solo tiene un punto!) En cualquier caso, irá seleccionando los puntos contenidos en la planificación y guardando los datos temporalmente en `/tmpSSD/SSDtmp0.txt`, a continuación de cada punto lo concatena en el archivo que se definió en la variable `ConcatenarSSD` (`SSDFile.txt`), al acabar la ejecución del script quedarán en el archivo `/tmpSSD/SSDFile.txt` los datos necesarios de los puntos.

Volvemos a `ExportarSSD.Script` y en la línea 30 llamamos al script de unix `buildSSD2Script` situado en `/tmpSSD` que, similar al script anterior de unix, filtra el archivo `FileSSD.txt` volcando los datos útiles (las distancias SSD) al archivo `ListaSSD.txt` situado en `tmpSSD`, y que únicamente contiene las distancias SSD de los puntos ordenadas en el orden en el que estaban en el archivo de exportación.

```
1 #!/bin/csh
2 # Exportar las distancias SSD a cero grados
3 # APerezRozos Julio 2007
4 # Borra temporales y extrae los datos necesarios
5
6 set outScript = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/SSDExport.Script.auto"
7 set Import = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/PoiList.txt"
8 set Pfad = "/usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD"
9
10 grep String /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/SSDFile.txt | cut -f2 -d\" > /usr/local/adacnew/PinnacleSiteData/Scripts/MAVV/tmpSSD/ListaSSD.txt
11
12 rm $Pfad/SSD*
13 rm $Import
14
```

Se vuelve al script `ExportSSD.Script` que en las líneas 33-37 copia el archivo `ListaSSD.txt` al directorio final de exportacion (`home/pinnamalaga/PRESCRIPCIONES`) y le cambia el nombre a `"SSD"+número de historia+".txt"`, y concluye cerrando las ventanas de configuración de campos de simulación en pinnacle (línea 39).

La ejecución continúa en la línea 18 de `ExportIMRT.Script`:

Se van a guardar *todos* los datos del trial, en este archivo está contenido el trial en su totalidad y de él se podría sacar cualquier parámetro necesario. Esto se hizo por motivos históricos para la extracción de algunos datos, pero ahora mismo podría ser algo prescindible. Al guardar este objeto se guardan también los archivos binarios asociados (que no se usarán y ocupan bastante espacio) así que a continuación se borran (líneas 25 a 29).

Notas:

- Falta por desarrollar y explicar la hoja de cálculo que importa y analiza todos estos datos
- faltaría por desarrollar y modificar los scripts para poder ser usados e instalados sin tener que añadir configuraciones particulares.
- ...

Málaga, 11 de noviembre de 2017
Alberto Pérez