

TCD_LAUNCHDLL

Dll que se encarga de ejecutar otras DLLs

Funciona desde dentro de DIGI:

Manera de ejecutarlo:

Primero hay que cargar la DLL: carga_dll = Fichero DLL

Ejemplo: carga_dll=D:\Usuarios\marcos\proyectos\TCD_LAUNCHDLL\Debug\TCD_LAUNCHDLL.dll

Luego ejecutar con: dll=Ejecutar

Para descargar: descarga_dll = Fichero DLL

descarga_dll=D:\Usuarios\marcos\proyectos\DIGI\Interactivos\TCD_LAUNCHDLL\Debug\TCD_LAUNCHDLL.dll

Descripción:

Aplicación que permite lanzar aplicaciones DLL dentro del entorno DIGI

EL programa lee de un fichero de configuración TCD_LaunchDLL.INI que debe estar en el mismo directorio donde esta la DLL.

Formato del fichero

// Si ponemos QC=S el programa no recargará el fichero BIN

// Si se pone QC=N el programa recargará el fichero BIN y lanzara la DLL con el parámetro /REPAIR

QC=S

// Catalogo

ECF=D:\Usuarios\marcos__kk\DIGIc1000R.ecf

// MALLA

MALLA=D:\Usuarios\marcos\mallA.bin

// Directorio de errores

DIR_ERR=D:\Usuarios\marcos__kk\DIGI

// DLLs a ejecutar

DLL=D:\Usuarios\marcos\proyectos\qg_programs\TCGR_DUP\Debug\TCGR_DUP.DLL

DLL=D:\Usuarios\marcos\proyectos\qg_programs\TCGR_LAZOS\Debug\TCGR_LAZOS.DLL

Si se pone QC=S el programa no recargara el fichero, si se pone QC=N el programa recargara el fichero y además lanzara todas las DLLs con el parámetro **/REPAIR** que hace que funcione en modo edición, es decir, arreglando los errores. Ej: TCGR2_DUP

Si se lanza los programas TCZ_MDT o TCGR_SENTIDO se les pasara como parámetros el fichero nombre del fichero BIN pero con extensión MDT

Las líneas que tienen // las ignora, el orden de los parámetros debe ser siempre el mismo (QC, ECF, DIR_ERR y DLLs a ejecutar.

Cada vez que se lanza genera un fichero BIN con los errores con el mismo nombre que el fichero pero añadiéndole _ERR.

Si se genera algún error automáticamente pone el fichero de errores de referencia

TCD_RECINTOS

Depuración de recintos creados en el entorno DIGI

Funciona desde dentro de DIGI:

Manera de ejecutarlo:

Primero hay que cargar la DLL: carga_dll = Fichero DLL

Ejemplo: carga_dll=D:\Usuarios\marcos\proyectos\TCD_Recintos\Debug\TCD_Recintos.dll

Ejecutar comando: dll=COMANDO

Comandos posibles

MakePolygons → Hace recintos con los elementos visibles

QCMakePolygons → Hace recintos y chequea errores con los elementos vis ibles.

MakeAllPolygons [FEA]→ Hace todos los recintos de cada IDC_R, solo visualiza los que tienen centroides.

MakeAllPolygonsSave [FEA] [/DGN]→ Hace todos los recintos de cada IDC_R, solo graba los que tienen centroides y los graba en un fichero que se llama igual que el fichero BIN pero se le añade “_rec”. Los salva con la zeta del centroide si el tipo 3D es calculado.

La opción **[FEA]** significa que los recintos se harán sólo con las líneas de borde asociadas

La opción **[DGN]** significa que también se grabara en formato DGN, con el mismo nombre del bin pero con extensión DGN. Usara el fichero SEED_DGN.INI

MakeIDCRPolygons [FEA]→ Hace todos los recintos de los IDC_Rs dados.

MakeIDCRPolygonsSave [FEA]] [/DGN] → Hace todos los recintos de los IDC_Rs dados., solo graba los que tienen centroides y los graba en un fichero que se llama igual que el fichero BIN pero se le añade “_rec”. Los salva con la zeta del centroide si el tipo 3D es calculado.

Ambos comandos admiten parámetro con los IDCRs sino se pasa pedira un IDCR

Ejemplo: MakeIDCRPolygons 123 124 125, deben estar separados por espacio

MakeIDCRPolygons 123 124 125 /FEA, Idem anterior pero los recintos se hacen con el borde

La opción **[DGN]** significa que también se grabara en formato DGN, con el mismo nombre del bin pero con extensión DGN. Usara el fichero SEED_DGN.INI

ShowOpenElements → Muestra los elementos lineales “abiertos” que no se usan para hacer recintos. Los muestra con código 92

MarkOpenElements → Marca con la feature 999 los elementos lineales que no se usan para hacer recintos.

ShowOpenNodeFromIDCR → Muestra los nodos abiertos con código 92. Las líneas se cogen por feature. Se usa la feature de cada IDCr

Este comando admite parámetro con los IDCRs sino se pasa pedirá un IDCR

Ejemplo: ShowOpenNodeFromIDCR 123 124 125, deben estar separados por espacio

AsignarBorde → Asigna la feature de borde a los elementos que forman parte de recinto.

El comando **AsignarBorde** asigna a cada línea formante de un recinto el borde asociado al recinto. Si una línea no puede tener asignado un borde por catálogo se produce un error, los errores de este tipo son una copia de la línea que no puede tener dicha asignación con código 90. Si se producen errores de topología en la carga de centroides el programa avisa y no asigna la feature. Entonces hay que ejecutar el comando **QCMakePolygons** para corregir los errores

VerManzanas → Calcula y visualiza las manzanas del fichero, usa el fichero manzana.ini. Se representan con el código 300. Para crear las manzanas elige los elementos que tienen el código dentro de manzana.ini a no ser que tenga la feature 997 (Es el caso de borde de zona cartografiada) o el borde de isleta de vial .

CrearManzanas → Genera centroide de manzana y asigna bordes. El código de la manzana es el literal del centroide de Recinto suelo urbano mas 300. Sino hay RSU pero hay etiquetas de uso rústico el código será 359 (59 + 300 Suelo no urbano) y si tampoco hay será para la visualización código 90 y los centroides se grabarán sin ocurrencia. Si hay etiqueta de suelo rústico y hay etiqueta de suelo urbano, a la etiqueta de suelo rústico se genera un centroide de suelo rústico (59), se genera sólo un centroide aunque haya mas de una etiqueta.

AsignaCruce → El comando hace todos los recintos por feature de calle (270, 273, 276 y 277). Localiza todos los recintos que son cruce (Su literal es "-1") y le asigna tantas ocurrencias como calles lleguen a dicho cruce

ViewByOcu → Visualiza el IDCR pedido haciendo un temático por ocurrencia. Utiliza el fichero ViewByOcu.INI.

Si el IDCR dado esta en el fichero INI pondrá los códigos definidos en el fichero pero sino encuentra alguna ocurrencia pondrá el código 399

Si el IDCR no esta en el fichero INI empezará en el código 300 y acabando en el 398. Si el recinto no tiene asignación se usará el código 399. Se puede pasar al comando directamente el IDCR a procesar. Ejemplo: ViewByOcu 230

VerRecintos → Activa desactiva la visualización de recintos

VerErrores → Activa desactiva la visualización de errores

VerEnvolventes → Activa desactiva la visualización de las envolventes

VerIslas → Activa desactiva la visualización de las islas

Para descargar: descarga_dll = Fichero DLL

descarga_dll=D:\Usuarios\marcos\proyectos\DIG\Interactivos\TCD_Recintos\Debug\TCD_Recintos.dll

EL programa lee de un fichero de configuración TCD_Recintos.INI que debe estar en el mismo directorio donde esta la DLL.

Formato del fichero

// Catalogo

ECF=D:\Usuarios\marcos_kk\DIG\lc1000R.ecf

Además busca el fichero manzana.ini

Formato:

on=-1699-1353-1241-1242-1243-1244-1245-1246-1247-1248-1249-1250-1403-1014-1006-1007-1008-1009-

Es importante que este el "-" después del igual y al final

Formato del fichero ViewByOcu.INI

// IDCR = 122 Masa arbolada

122;90;300

122;91;301

122;92;302

122;93;303

// IDCR = 126 Suelo Urbano

126;50;310

126;51;311

Formato del fichero SEED_DGN.INI

// Seed DGN

ECF=d:\data\seed.dgn

INCX=0

INCY=4000000

INCZ=0

Descripción:

Aplicación que permite generar los recintos posibles con las líneas activas en ese momento y que además chequea que cada recinto tiene su Centroid:

Por el chequeo que hace puede marcar 3 tipos de error:

61657 → Polígono sin Centroides, marca el polígono entero

61658 → Centroides sin recinto, marca el Centroides

61660 → Centroides duplicados, marca el centroide

Tanto los errores como los recintos no se graban en el fichero BIN simplemente se visualizan.

Los recintos que se generar tienen códigos que van del 300 al 399, por lo que para que se vean los rellenos en la tabla de digi hay que indicárselo, lo mismo si se quiere rellenar el código 61657.

Ejemplo:

```
300 -1 8 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 "Relleno 300"
```

```
301 -2 8 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 "Relleno 301"
```

TCD_COMMANDS

Comandos para DIGI

Funciona desde dentro de DIGI:

Manera de ejecutarlo:

Primero hay que cargar la DLL: carga_dll = Fichero DLL

Ejemplo: carga_dll=D:\Usuarios\marcos\proyectos\TCD_TopProcess\Debug\TCD_Commands.dll

Ejecutar comando: dll=COMANDO

Comandos posibles

ZInterpoBynodes → *Cambia la zeta de una línea. El comando va pidiendo puntos de una entidad hasta que se le da el origen y el fin de dicha entidad.*

Si se cambia de entidad se corta el comando

Con los puntos dados hace una interpolación lineal

MostrarCoordenadas → *Displaya las coordenadas con código 102 y los nodos con código 101*

MostrarCoordenadasYA → *Displaya las coordenadas con código 102 y los nodos con código 101, sólo se muestra una vez que le das al clic*

Boratrib → *Borra asignaciones de feature dada, si se pone * borra todas las asignaciones de feature. Se pide una feature, y entonces el programa elimina todos los elementos que tienen la feature pedida. No quitara la feature que corresponde al código*

CargaMDT → *Carga fichero MDT. Carga el fichero MDT asociado al fichero bin, es decir, con el mismo nombre pero extensión MDT. El código que se usa para dibujar el fichero MDT es el 91*

ObtenerZMDT → *Da la zeta del mdt asociado*

ShowMDT → *Muestra / No muestra el fichero MDT*

DesCargaMDT → *DesCarga fichero MDT*

InsertaTxtMDT → *Introduce un texto que se coloca con la Zeta del MDT, pone el código del elemento activo y el tamaño y la justificación que le corresponde según catalogo*

InfoRecintos → *Obtiene los recintos que puede formar una línea*

InfoComponentes → *Obtiene las líneas que pueden formar un recinto, para ver el recinto*

VerPorObjeto → *Activa los códigos de un objeto dado por el usuario*

VerPorIDCR → *Activa los códigos de un objeto dado por el usuario*

ToNoError → *Para marcar los errores de los programas como no error, Paso al nivel "62"*

Asigpo → *Asignación masiva de portales. Pide un elemento con asignación a la tabla 130, 131 o 132, pide un elemento que sea un recinto y todos los portales que estén dentro del recinto asigna la ocurrencia del primer elemento dado. Si un portal apuntaba a otra ocurrencia lo actualiza*

EL programa lee de un fichero de configuración TCD_Commands.INI que debe estar en el mismo directorio donde esta la DLL.

Formato del fichero

// Catalogo
ECF=D:\Usuarios\marcos__kk\DIGlc1000R.ecf
Para descargar: descarga_dll = Fichero DLL
descarga_dll= D:\Usuarios\marcos\proyectos\TCD_TopProcess\Debug\TCD_Commands.dll

Si esta activado el comando repite de DIGI el comando se ejecutara hasta que apretemos el ESCAPE

TCD_Path

Comandos para DIGI basados en la topología lineal

Funciona desde dentro de DIGI:

Manera de ejecutarlo:

Primero hay que cargar la DLL: carga_dll = Fichero DLL

Ejemplo: carga_dll=D:\Usuarios\marcos\proyectos\TCD_TopProcess\Debug\ TCD_Path.dll

Ejecutar comando: dll=COMANDO

Comandos posibles

AsigOcuPath → Asigna ocurrencia a los elementos que forman parte de un PATH.

El comando elimina todos los nodos superfluos (en memoria) que se tienen en los elementos visibles. Una vez eliminados todos esos nodos revisa la asignación de cada tramo y si es correcto asignará los tramos que no tienen asignación con la correcta.

Dibuja 2 tipos de errores

- Código 93: Elemento completo sin asignar
- Código 94: Elemento completo con 2 asignaciones

TCD_TOPPROCES

Procesos basados en la topología

Funciona desde dentro de DIGI:

Manera de ejecutarlo:

Primero hay que cargar la DLL: carga_dll = Fichero DLL

Ejemplo: carga_dll=D:\Usuarios\marcos\proyectos\TCD_TopProcess\Debug\TCD_TopProcess.dll

Ejecutar comando: dll=COMANDO

Comandos posibles

LineToCota → Sube las líneas de los recintos a la zeta mas alta de las cotas adyacentes

ElevaZCalc → Modifica la z de los elementos definidos en el catalogo. La configuración se lee de la columna "geo_elevazcalc". Hará los recintos en memoria y solo usará los recintos que tengan su centroide

Para descargar: descarga_dll = Fichero DLL

descarga_dll= D:\Usuarios\marcos\proyectos\TCD_TopProcess\Debug\TCD_TopProcess.dll

EL programa lee de un fichero de configuración TCD_TOPPROCESS.INI que debe estar en el mismo directorio donde esta la DLL.

Formato del fichero

// Catalogo

ECF=D:\Usuarios\marcos__kk\DIGIc1000R.ecf

Descripción:

LineToCota → Las líneas que forman parte del recinto se suben a la cota del centroide. Trabaja con los elementos visibles