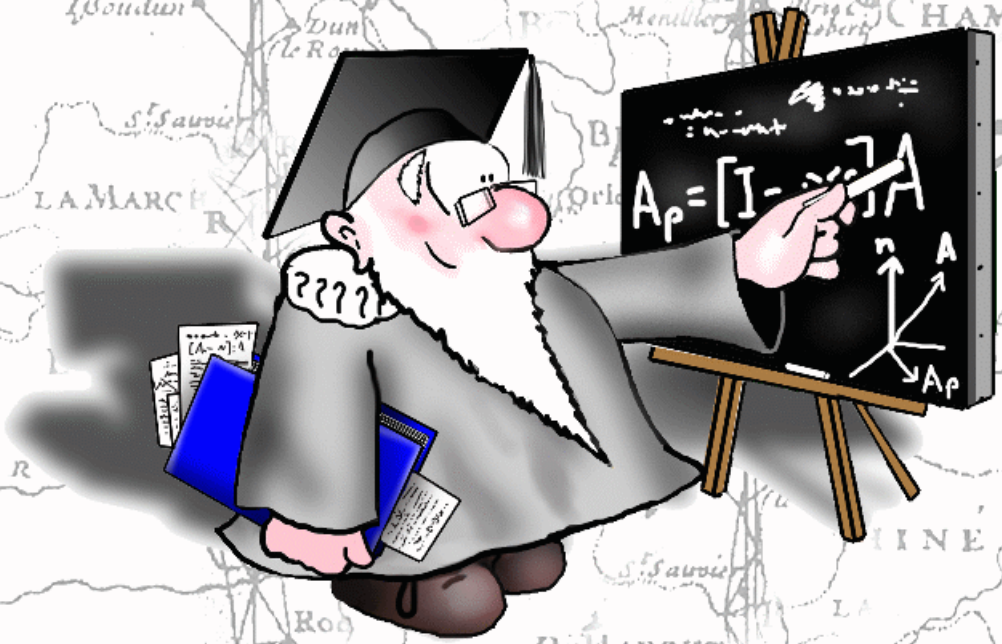
GUIONES SML

Tutoriales

Escribiendo Guiones con SML



en

TNTmips®

TNTedit™

TNTview®

Antes del Tutorial

Este folleto introduce técnicas para crear guiones en el Lenguaje de Manipulación Espacial (SML) en los productos TNT. Los ejercicios en este folleto introducen los conceptos y técnicas para escribir poderosos guiones para personalizar manipulaciones de los objetos de datos espaciales en sus Archivos de proyecto TNT.

Requisitos Previos Este folleto asume que usted a completado los ejercicios en los siguientes folletos *Tutoriales*: *Desplegando Datos Geoespaciales, y Navegando.* Por favor consulte esos folletos y el manual de referencia de TNTmips para cualquier revisión de las habilidades esenciales y técnicas básicas que necesite. Este folleto también asume que usted tiene al menos un conocimiento fundamental de uno o más lenguajes de programación tales como C, BASIC o Pascal. Usted puede empezar a usar SML aún si usted no tiene una formación en programación, pero SML es un lenguaje poderoso y permite obtener el máximo beneficio en las manos de un buen programador.

Datos de Ejemplo Los ejercicios presentados en este folleto utilizan datos de ejemplo distribuidos con los productos TNT. Si no tiene acceso al CD de productos TNT, usted puede bajar los datos desde el sitio web de MicroImages. En particular este folleto usa los guiones de la colección de datos LITEDATA / SML, los subdirectorios CUSTOM, MACRSCR y TOOLSCR bajo el directorio principal de TNT, y objetos en las colecciones de datos CB_DATA, SF_DATA, SURFMODL, y EDITRAST. Instale los archivos de ejemplo en su disco duro; usted podría encontrar problemas si trabaja directamente con los datos de ejemplo en el CD-ROM.

Mas Documentación Este folleto solo intenta ser una introducción al uso del Lenguaje de Manipulación Espacial. Para mayor información, consulte el manual de referencia de TNT y especialmente la referencia en línea de SML.

TNTmips y TNTlite™ TNTmips viene en dos versiones: la versión profesional y la versión libre TNTlite. Este folleto se refiere a las dos versiones como "TNTmips." Si usted no compra la versión profesional (la cual requiere de una llave de licencia de software), TNTmips opera en modo TNTlite, el cual limita el tamaño de sus materiales de proyecto y activa el compartir de datos únicamente con otras copias de TNTlite. SML no está disponible en TNTatlas. Todos los ejercicios pueden completarse en TNTlite utilizando los geodatos de ejemplo proporcionados.

Keith Ghormley and Randall B. Smith, Ph.D., 31 October 2001

Sin una copia a color de este folleto podría ser difícil identificar algunos puntos importantes en algunas ilustraciones. Usted puede imprimir o leer este folleto a color en el sitio Web de MicroImages. Este sitio Web es también su fuente de nuevos Tutoriales sobre otros temas. Usted puede descargar una guía de instalación, datos de ejemplo y la última versión de TNTlite.

http://www.microimages.com

SML en los Productos TNT

El Lenguaje de Manipulación Espacial (SML) es un lenguaje de programación que le permite escribir guiones que operan sobre objetos de datos geoespaciales en los Archivos de Proyecto de TNT. Los guiones SML se pueden ejecutar desde menúes personalizados, barras de iconos, iconos del escritorio de la computadora y aún desde la línea de comandos del sistema operativo.

SML es una herramienta de personalización y diseño. Con SML puede usar los productos TNT para tareas mas allá de los procesos predefinidos que se encuentran en los menúes estándar de TNT. Usted puede crear guiones para simples procesos o bien productos completos de propósitos especiales para objetivos de mercado privado. Usted puede aún empaquetar sus guiones conjuntamente con objetos de geodatos en un archivo de proyecto y distribuir todo el paquete como un proyecto llave en mano: APPLIDAT (APPLIcation plus Data).

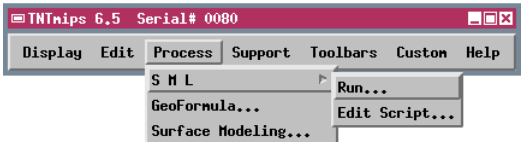
MicroImages proporciona varias piezas de ejemplos de guiones SML para darle modelos con los que puede trabajar. Usted puede examinar y adaptar los guiones que van desde simples rutinas de procesamiento a complejos APPLIDATs que contienen cientos de líneas de código. Sus guiones SML serán fáciles de modificar y mejorar a lo largo de la vida de su proyecto. Usted rápidamente puede hacer prototipos y probar las características del programa.

PASOS

☑ seleccione Instalar los Ejemplos de Guiones SML del CD-ROM de los productos TNT en el menú de instalación

En ejercicios posteriores usted accederá los guiones en los subdirectorios /CUSTOM bajo el directorio de los productos TNT.

Los ejercicios en las páginas 4-18 introducen los conceptos básicos de SML y las convenciones de los guiones. Las páginas 19-27 ilustran sobre técnicas de programación específicas para diferentes tipos de objetos de geodatos. El resto del folleto introduce sobre técnicas avanzadas de desarrollo de SML: creando ventanas de dialogo; guiones de animaciones; APPLIDATs; guiones de Herramientas y Macros.



SML en TNTmips

SML en un APPLIDAT

Los guiones SML son completamente independientes de la plataforma; estos se ejecutan sin modificación en cualquier computadora que corre los productos TNT.

SML está disponible en TNTmips, TNTedit, y TNTview. Adicionalmente todas las formas de guiones en los productos TNT usan estructuras delineadas de SML (con variaciones menores):

- guiones de consultas y estilos
- APPLIDATs
- GeoFormulas
- Guiones de Herramientas
- CartoScripts
- Guiones de Macros

Referirse a los folletos Tutoriales *Generando y Usando Consultas, Usando CartoScripts, y Usando Formulas Geoespaciales* para información acerca de los tipos particulares de guiones.

Ejecutar VIEWSHED.SML

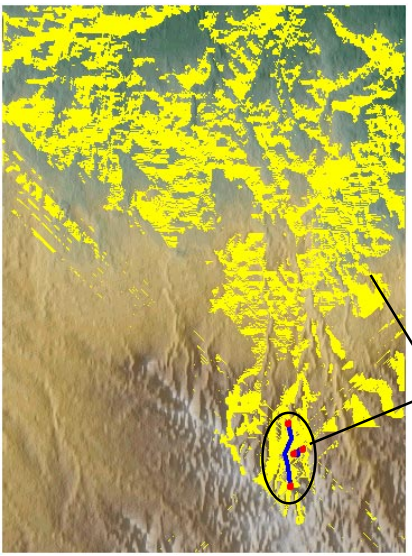
PASOS

- ☑ seleccione Process / SML / Edit Script
- ☑ escoja File / Open / *.SML File y seleccione VIEWSHED.SML de la carpeta LITEDATA / SML
- ☑ recorra por el guión para una primera mirada a SML
- ☑ clic [Run...] en el fondo de la ventana
- ☑ cuando sea requerido el ingreso del raster "RIN", seleccione ELEVATION del Archivo de Proyecto CB_TM en LITEDATA / CB_DATA
- ☑ para el ingreso del vector "V", seleccione VROAD del Archivo de Proyecto VIEWSHED en LITEDATA / SML
- ☑ seleccione un nuevo Archivo de Proyecto y objeto para el raster de salida "ROUT"
- ☑ utilice el proceso de despliegue para mirar los tres layers: CB_TM / ELEVATION, su nuevo raster de salida, y VIEWSHED / VROADS

El guión VIEWSHED.SML conjuntamente con sus datos de ejemplo en VIEWSHED.RVC están contenidos en el CD-ROM de los productos TNT, y también se hallan disponibles en la página WEB de MicroImages. El guión crea un objeto raster binario de salida que muestra cuales partes de la superficie de elevación de ingreso son visibles desde el flujo de puntos a lo largo de los elementos lineales de ingreso. Muchas aplicaciones que tratan con las características de las superficies de líneas de vista pueden usar las técnicas ilustradas en este guión.

Inicie SML y cargue el guión VIEWSHED.SML siguiendo los pasos listados. Antes de ejecutar el guión, recórralo y estudie su contenido. A menos que usted no tenga algún conocimiento con un lenguaje de programación como C o BASIC, usted podrá reconocer formas de declaración y estructuras de programación.

Note que el trabajo más pesado del guión es efectuado con llamadas a varia funciones SML, tales como `RasterToBinaryViewshed()`. MicroImages está constantemente añadiendo nuevas funciones a SML. Estando enterado de cuales funciones están disponibles y entendiendo que hacen es esencial para producir la mayoría de SML. Adicionalmente al uso de las funciones propias de SML, usted puede escribir en C sus propias extensiones SML con TNTsdk, e invocar programas externos desde el guión SML (ver página 28).



VIEWSHED.SML produce un raster binario (los 1's mostrados aquí en amarillo) que indica las áreas visibles en una superficie de elevación (mostradas en pseudo - color) desde un flujo de puntos a lo largo de los elementos lineales de un vector de entrada (mostrados en azul). Por lo tanto si los elementos lineales representan caminos, luego las áreas amarillas definen las vistas disponibles para los viajeros en la ruta.

Principios Para Elaborar Guiones

Un guión SML puede ser desde algo tan simple como una declaración hasta un largo programa estructurado con ramificaciones de estructuras lógicas anidadas y llamadas a programas externos. Existen unas pocas restricciones estructurales formales.

Usted puede usar espacios en blanco en casi cualquier forma. SML no se preocupa si usa espacios o tabulaciones (tab) para las sangrías de las líneas, o si deja líneas en blanco, o aún si rompe una declaración en la mitad y la continúa en la siguiente línea. Consecuentemente VIEWSHED.SML tiene una llamada a una función que esta rota en la mitad de la lista de argumentos y continúa en la siguiente línea:

```
MapToObject( georefR, xVector,Vector,
       Rin, rCol, rLine )
```

PASOS

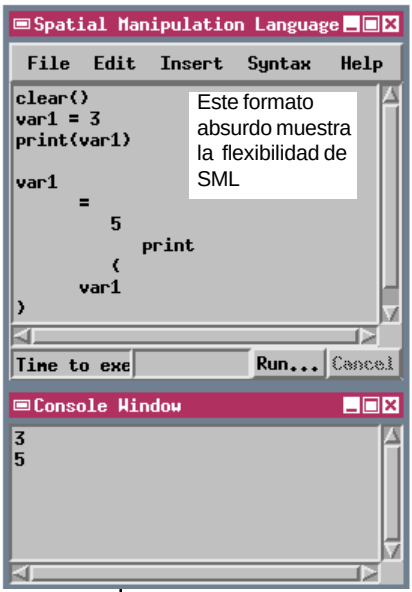
- ☑ despeje la ventana SML seleccionando New del menú File
- ☑ escriba el guión ilustrado abajo, usando tab o la barra espaciadora para la sangría del texto
- ☑ seleccione Syntax / Check para verificar la sintaxis
- ☑ clic [Run] para ejecutar el guión

La **Ventana SML** es un simple editor de texto que proporciona acceso a una lista de funciones y verificación de la sintaxis.

De la misma manera, las declaraciones pueden continuar a lo largo de varias líneas. Seleccione New del menú File para despejar la ventana SML. Luego escriba el guión ilustrado aquí. La declaración inicial `clear()` borra el contenido de la ventana de Consola. La declaración `var1 = 3` asigna el valor de 3 a la variable `var1`. La primera declaración `print()` reproduce el valor de `var1` a la ventana de la Consola.

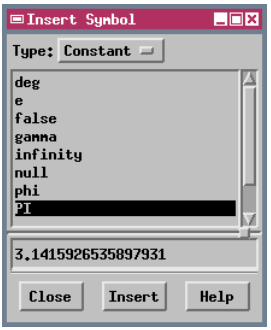
Note que las siguientes tres líneas contienen una sola declaración: `var1 = 5`. La declaración final `print()` es así mismo distribuida a lo largo de cuatro líneas. Clic [Run...] para ejecutar el guión.

Por supuesto que el absurdo formato en este ejemplo se proporciona únicamente para ilustrar lo flexible del formato que soporta SML. En sus propios guiones, utilice sangrías, espacios y líneas en blanco para facilitar la legibilidad y reflejar la estructura lógica del programa. Si usted llega a usar un formato o sintaxis ilegal, SML no ejecuta el guión cuando hace clic en [Run], pero anuncia un mensaje de error y coloca el cursor en el guión en el punto del error.



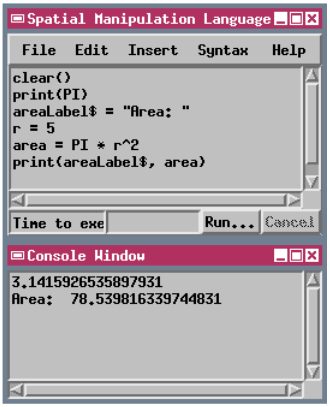
La **Ventana de la Consola** muestra los resultados de print() y otros textos de operaciones de input / output

Variables y Constantes



PASOS

- ☑ seleccionar File / New para despejar la ventana SML
- ☑ escriba en las dos primeras líneas del guión en la forma
  clear()
  print()
  y deje su cursor entre los paréntesis de la declaración print()
- ☑ seleccione Insert / Symbol, escoja pi de la lista, y clic [Insert], [Close]
- ☑ escriba el resto del guión indicado abajo y clic [Run]



Variables pueden usarse para entidades alfanuméricas, numéricas, lógicas, arreglos, clases y objetos (CAD, raster, vector, regiones, y TIN). Las variables son creadas cuando el guión las menciona por primera vez. Con la excepción de los Arreglos y Clases, ellas no necesitan que sean declaradas por adelantado. Los nombres pueden ser hasta de 100 caracteres de largo y siguen las siguientes convenciones:

String: la letra inicial es en minúscula; debe terminar con el signo '$'. En el guión de ejemplo de esta página, areaLabel$ es una variable string.

Numeric: la letra inicial es minúscula; no puede terminar en '$'. En el guión de ejemplo de esta página, r y area son variables numéricas.

Object: la letra inicial es mayúscula.

Ejemplo GetInputRaster(R)

Logical: implementada como numérica donde 0 = falso, y todos los valores distintos de cero = verdadero. Luego:

```
done = 0;
if (condition) done = 1;
if (done) <statement>;
```

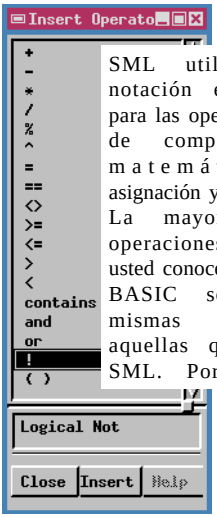
Los nombres de **Array** y **Class** siguen las convenciones de las variables string y numéricas. Usted debe declarar un arreglo o clase antes de usarlos. Encierre el índice de un arreglo en paréntesis rectos:

```
array numlist[10];
class COLOR red;
numlist[1] = 256;
```

Usted puede usar la ventana de inserción de símbolos (Insert / Symbol) para insertar las variables previamente utilizadas en el guión, o para insertar *constantes* predefinidas (cuyos valores no pueden ser cambiados). Utilice la opción del menú Type para escoger el tipo de variable (o constante) y mirar la lista asociada. Insertando nombres de variables en lugar de escribirlos puede reducir los errores de escritura. Sus nombres de variables no aparecen es estas listas hasta que usted use la operación de verificación de sintaxis (vea página 16) o ejecuta el guión.

Expresiones y Declaraciones

Expresiones son estructuras que se reducen a un valor. Así pi^2 , 5.10, y R[i,j]/100 todas son expresiones. Las expresiones pueden usarse en el lado derecho de las declaraciones de asignación y como argumentos en llamadas a funciones.



SML utiliza la notación estándar para las operaciones de comparación, matemáticas, asignación y lógicas. La mayoría de operaciones que usted conoce de C o BASIC son las mismas como aquellas que usa SML. Por favor

Declaraciones pueden ser simples o complejas. Una declaración simple puede consistir de una asignación, tal como

```
area = pi * r^2;
```

Una *declaración compleja* esta delimitada por las palabras “begin” y “end” en la forma

```
if (condition) begin
    function(r);
    area = pi * r^2;
end
```

SML también le permite usar corchetes (“corchetes curvos”) en lugar de escribir “begin” y “end”:

```
if (condition) {
    function(r);
    area = pi * r^2;
}
```

Las *declaraciones condicionales* tienen la forma

```
if (<condition>) then <statement>
else <statement>;
```

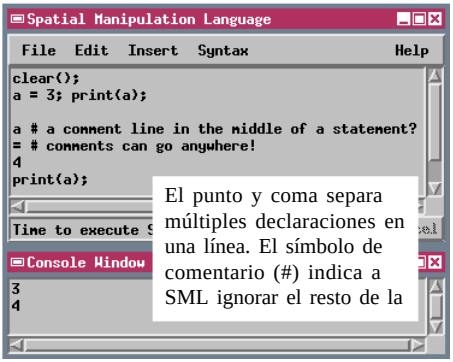
La cláusula *else* es opcional, tanto como el “*then*”:

```
if (<condition>) <statement>;
```

PASOS

- ☑ seleccione File / Open / *.SML File y luego escoja LITEDATA / SML / EXPRESS.SML
- ☑ ejecute el guión
- ☑ cambie el umbral del area para la condición if y ejecute el guión nuevamente

Es una buena práctica (aunque opcional) el usar el signo terminal (el punto y coma, “;”) para marcar el final de una declaración. Usando el signo terminal también le permite colocar múltiples declaraciones en una línea, separándolas con puntos y coma.



El punto y coma separa múltiples declaraciones en una línea. El símbolo de comentario (#) indica a SML ignorar el resto de la

El signo de comentario (“#”) indica a SML que ignore el resto de la línea. Si un signo de comentario es el primero en una línea, SML ignora toda la línea. Usted puede también colocar un comentario en la misma línea con otros símbolos SML, con tal de que los símbolos estén antes del comentario.

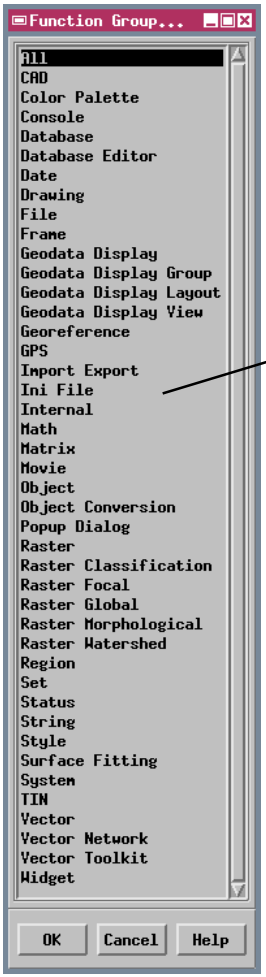
Funciones Internas

PASOS

- ☑ despeje la ventana SML con File / New
- ☑ seleccione Insert / Function
- ☑ clic el botón Function Group y examine la librería de funciones para cada categoría

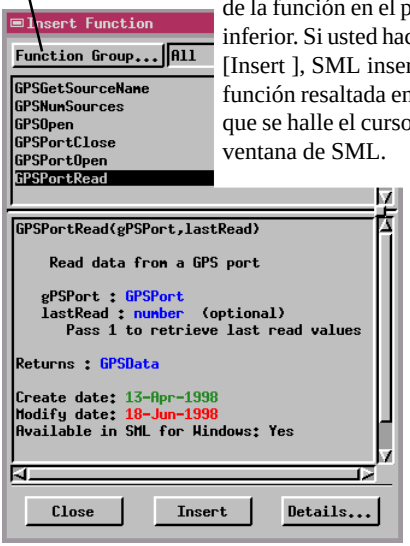
El poder real de SML está en su rica librería de funciones que le permiten crear, leer y escribir objetos y subobjetos geoespaciales en sus Archivos de Proyecto TNT. Funciones matemáticas estándar se incluyen conjuntamente con funciones especializadas de despliegue, interface y manipulación de datos. MicroImages constantemente está mejorando y expandiendo las funciones SML para darle a usted más formas de trabajar con sus datos geoespaciales.

Seleccione Insert / Funtion para abrir la ventana de inserción de funciones, la cual le permite seleccionar funciones y mirar su uso y especificaciones de formato. Clic el botón Function Group para examinar las funciones disponibles de cada categoría. Conforme usted se desplaza a lo largo de la lista de funciones, la definición en el panel inferior cambia para mostrar la función actual. Clic el botón Insert para copiar la función en la ventana de guiones SML.



El botón Funtion Group abre una lista deslizable de las categorías de funciones

La ventana de Inserción de Funciones ofrece una lista deslizable de las funciones en el panel superior, y una definición de la función en el panel inferior. Si usted hace clic [Insert], SML inserta la función resaltada en la posición que se halle el cursor en la ventana de SML.



Ayuda de Funciones en Línea

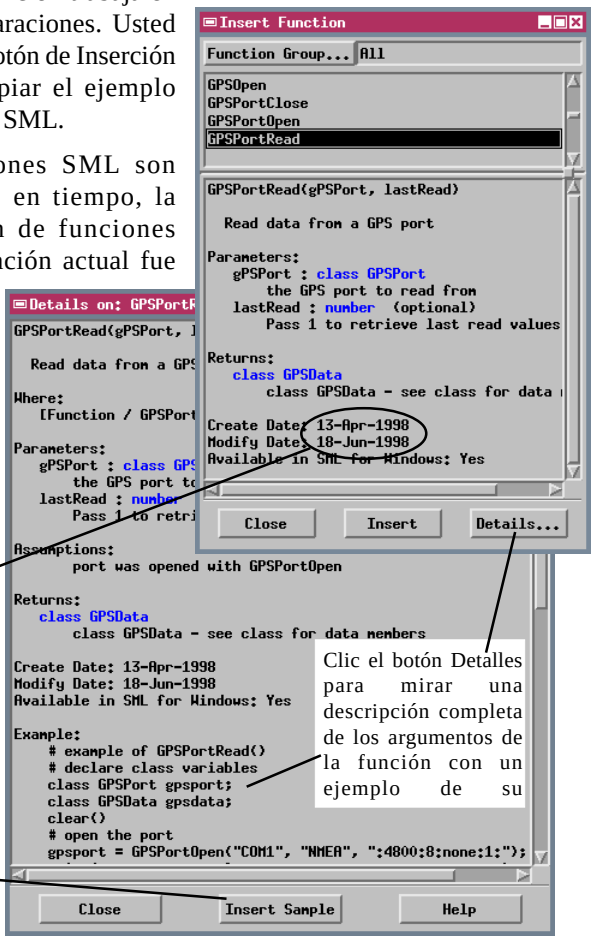
La documentación de soporte para las funciones SML está incorporada en el proceso. Primero el panel inferior de la ventana de Inserción de Función provee de una definición simple, mostrando cada argumento y su tipo de dato. Usted puede hacer clic en el botón Insert para copiar una instancia completa de la función en la ventana de SML

Para mayor información, clic en el botón Details en la ventana de Inserción de Funciones. SML abre la ventana de Detalles que proporciona detalles completos, más una sección de trabajo de código que muestra como la función trabaja en una secuencia de declaraciones. Usted puede hacer clic en el botón de Inserción de Ejemplos, para copiar el ejemplo completo en la ventana SML.

Dado que las funciones SML son mejoradas de tiempo en tiempo, la ventana de inserción de funciones muestra cuando la función actual fue actualizada por última vez. Este pendiente de las modificaciones que proveen nuevas capacidades opcionales de funciones que haya usado.

PASOS

- ☑ seleccione All en el recuadro de texto Function Group
- ☑ desplazar hasta la función GPSPortRead()
- ☑ clic el botón Details
- ☑ clic [Insert Sample]
- ☑ examine las líneas del guión recientemente insertado en la ventana de guiones SML



La fecha de creación indica cuando la función fue introducida a SML. La fecha de Modificación indica cuando la función fue actualizada por última vez. Algunas veces los argumentos opcionales son añadidos a una función para expandir sus capacidades.

Clic el botón Detalles para mirar una descripción completa de los argumentos de la función con un ejemplo de su

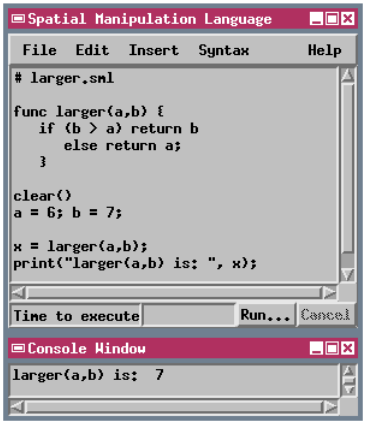
Clic Insertar Ejemplo para copiar toda la sección de ejemplo del código en la ventana SML

- ☑ cierre las ventanas Details e Insert Function cuando haya completado este ejercicio

Funciones Definidas por el Usuario y Procedimientos

PASOS

- ☑ seleccione File / Open / *.SML File y abra LARGER.SML de la carpeta LITEDATA / SML
- ☑ ejecute el guión



SML permite definir sus propias funciones y procedimientos que puede usar para encapsular secuencias de pasos de programa que deben repetirse en varios lugares del guión. Las funciones definidas por el usuario deben retornar un valor, mientras que los procedimientos no. Por supuesto usted debe declarar una función o procedimiento antes de invocarlo, utilizando la forma:

```
func funcname ([parmlist])
    { statement; statement; ...
                return expr }
proc procname ([parmlist])
    { statement; statement; ... }
```

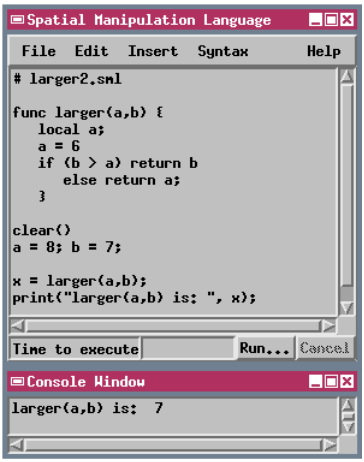
este ejemplo simple usado en este ejercicio encuentra el mayor de dos valores.

A menos que se declare lo contrario, todas las variables del guión son globales. Esto significa que sus funciones y procedimientos pueden usar y modificar las variables definidas en cualquier parte del guión. Cualquier variable global y Clase usada en funciones y procedimientos deben ser declaradas antes de las definiciones de las funciones. En un guión largo o complejo, el alcance de estas variables globales podría causar consecuencias imprevistas. Para limitar el alcance de una variable a una función en particular o procedimiento, usted debe declarar la variable como una variable local dentro de la definición de la función:

```
local x;
```

en donde x es un nombre de variable. Las variables locales pueden tener el mismo nombre de las variables globales en cualquier parte del guión, no obstante esta no es una práctica recomendada. El guión LARGER2 declara una variable local “a” dentro de la definición de la función `larger(a,b)` y le asigna un valor de 6. En la parte principal del guión, es declarada una variable global con el mismo nombre y asignado el valor de 8. Como lo ilustra el resultado del guión, la variable global es ignorada y el valor de la local es utilizado en su lugar por la función.

- ☑ seleccione File / Open / *.SML File y abra LARGER2.SML de la carpeta LITEDATA / SML
- ☑ ejecute el guión



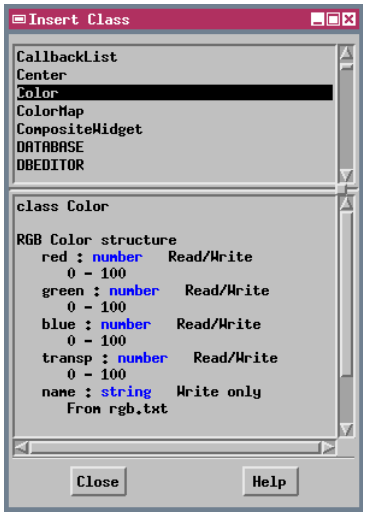
Usando Clases

Una Clase es una variable compleja que consiste de múltiples miembros, de la misma forma que un registro de una base de datos consiste de múltiples campos. Una variable de clase puede tener cualquier número de miembros y los miembros pueden ser cualquier tipo de datos, incluyendo otras clases.

PASOS

- ☑ despejar la ventana SML con File / New
- ☑ seleccione Insert / Class
- ☑ deslizar por la lista en el panel superior de la ventana Insert Class y seleccione la clase Color

Las variables de clase están diseñadas para pasar información desde y hacia funciones complejas. En muchos casos, los miembros de una variable de clase se fijan únicamente por una llamada a una función, y por lo tanto son solo de lectura desde el punto de vista del guión, a ellos no se puede asignar nuevos valores por medio de declaraciones de asignación.



Una clase debe ser declarada con la palabra clave de Clase, de la forma:

```
class COLOR background
```

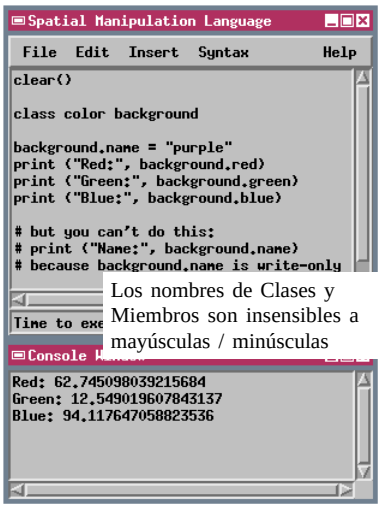
la cual declara a "background" como una variable de Clase de tipo Color. Los miembros de una clase son especificados en la forma name.member (de igual forma como los valores de base de datos son especificados de la forma table.field). Por ejemplo, la clase color tiene cinco miembros a los que pueden ser asignados valores con declaraciones de la forma:

```
background.red=50
background.green=75
background.blue=20
background.transp=0
```

El nombre del miembro de la Clase Color, es utilizado únicamente para pasar los valores de red, green y blue a la variable de clase desde el archivo de referencia estándar RGB.TXT. En consecuencia:

```
background.name = "purple"
```

fija los componentes RGB de la variable de clase "background" de acuerdo a la definición de "purple" en RGB.TXT. El nombre del miembro es solo de escritura y no puede ser leída en otra parte del guión.



Los nombres de Clases y Miembros son insensibles a mayúsculas / minúsculas

Herencia de los Miembros y Chequeo de Tipos

PASOS

- ☑ seleccione Insert / Class
- ☑ seleccione POINT2D en el panel superior de la ventana Insert Class, y examine sus miembros
- ☑ seleccione POINT3D en el panel superior de la ventana Insert Class, y examine sus miembros
- ☑ seleccione XmPushButton en el panel superior de la ventana Insert Class
- ☑ siga la línea de Clase y derivación de Miembros mostrada en el panel inferior

Series de Clases derivadas como aquellas mostradas abajo son utilizadas en SML para representar las estructuras de X Windows / Motif usadas para crear la interface de windows. Todos los componentes de la interface son categorías de un componente básico llamado un *widget*

Un importante concepto con las Clases es la *herencia*. La Clase POINT2D representa la ubicación de un punto 2-dimensional; sus miembros son las coordenadas x, y del punto. LA Clase POINT3D se dice que es *derivada* de la clase POINT2D. Esto significa que una variable de clase que usted declara como POINT3D no solamente tiene su propio miembro z, pero también *hereda* los miembros x, y de la clase POINT2D. Usted puede utilizar los miembros heredados de una Clase de la misma forma que los haría con los miembros nativos.

El uso de Clases le permite una *estricta verificación de tipos*. De forma que cuando invoca una función que requiere un POINT2D para parámetro, usted puede pasar cualquier POINT2D (o una clase derivada). Pero la función rechazará cualquier variable que no sea un POINT2D. Por ejemplo, usted no puede pasar a tal función una Clase Color, porque Color no es un POINT2D. Por contraste dado que POINT3D es derivado de POINT2D, usted puede pasar un POINT3D o cualquier derivado de POINT2D a una función que requiera un POINT2D.

Las definiciones de XmLabel y XmPushButton muestran la herencia compartida.

```
class XmPushButton

Simple push button widget
   ActivateCallback : class XmCallbackList   Read only
   ArmCallback : class XmCallbackList   Read only
   DefaultButtonShadowThickness : number   Read/Wri
   FillOnArm : number   Read/Write
   ShowAsDefault : number   Read/Write
   MultiClick : string   Read/Write
      Possible values:
         "MULTICLICK_DISCARD"
         "MULTICLICK_KEEP"

class XmPushButton is derived from class XmLabel
and inherits the following members from it

   Alignment : string   Read/Write
      Possible values:
         "ALIGNMENT_BEGINNING"
         "ALIGNMENT_CENTER"
         "ALIGNMENT_END"
   LabelString : string   Read/Write
   MarginBottom : number   Read/Write
   MarginHeight : number   Read/Write
   MarginLeft : number   Read/Write
   MarginRight : number   Read/Write
   MarginTop : number   Read/Write
   MarginWidth : number   Read/Write
   RecomputeSize : number   Read/Write
```

```
class XmLabel

A static text label; also base class of buttons
   Alignment : string   Read/Write
      Possible values:
         "ALIGNMENT_BEGINNING"
         "ALIGNMENT_CENTER"
         "ALIGNMENT_END"
   LabelString : string   Read/Write
   MarginBottom : number   Read/Write
   MarginHeight : number   Read/Write
   MarginLeft : number   Read/Write
   MarginRight : number   Read/Write
   MarginTop : number   Read/Write
   MarginWidth : number   Read/Write
   RecomputeSize : number   Read/Write

class XmLabel is derived from class XmPrimitive
and inherits the following members from it

   BottomShadowPixel : number   Read/Write
   BottomShadowColor : class COLOR   Write only
   ForegroundPixel : number   Read/Write
   ForegroundColor : class COLOR   Write only
   HighlightPixel : number   Read/Write
   HighlightColor : class COLOR   Write only
   HighlightOnEnter : number   Read/Write
   HighlightThickness : number   Read/Write
```

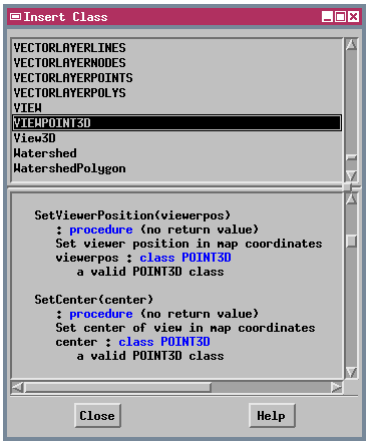
Siga la herencia desde Widget a XmPrimitive a XmLabel a XmPushButon

Métodos de Clases

Algunas Clases incluyen sus propias funciones y procedimientos, las cuales son llamadas colectivamente Métodos de Clases. Los métodos de clases pueden ser utilizados para pasar valores dentro de una clase o para ejecutar algunas otras operaciones relacionadas a la clase. Los métodos de clase son invocados usando la forma name.method(), donde name es el nombre de la variable de clase.

PASOS

- ☑ seleccione Insert / Class
- ☑ seleccione VIEWPOINT3D en el panel superior de la ventana Insert Class
- ☑ desplace el panel inferior y examine los métodos de clase



La clase VIEWPOINT3D representa los parámetros fijados para una reproducción 3D en la ventana de Vista 3D. Esta incluye los miembros ViewPos, una variable de clase POINT3D que contiene las coordenadas x,y,z de la ventana de inspección. Un método de clase es utilizado para pasar los valores requeridos dentro de la clase:

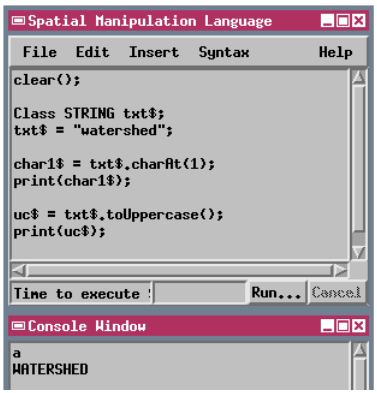
```
Class VIEWPOINT3D vp;
Class POINT3D vpos;
vpos.x = 523487;
vpos.y = 1473245;
vpos.z = 2000;
vp.SetViewerPosition(vpos);
```

Este método de clase es un procedimiento, y por lo tanto no devuelve un valor.

- ☑ seleccione STRING en el panel superior de la ventana Insert Class
- ☑ desplace el panel inferior y examine los métodos de clase

Los métodos en la Clase STRING son todas funciones que devuelven bien sea una cadena de caracteres o un valor numérico. Intente escribiendo y ejecutando el siguiente ejemplo:

```
clear();
Class STRING txt$;
txt$ = “watershed”;

char1$ = txt$.charAt(1);
print(char1$);

uc$ = txt$.toUppercase();
print(uc$);
```



El método CharAt(n) devuelve el carácter enésimo en la cadena de caracteres – string- (indexada a 0 con el carácter más hacia la izquierda). El método toUppercas() devuelve una copia de la cadena de caracteres todos en mayúsculas.

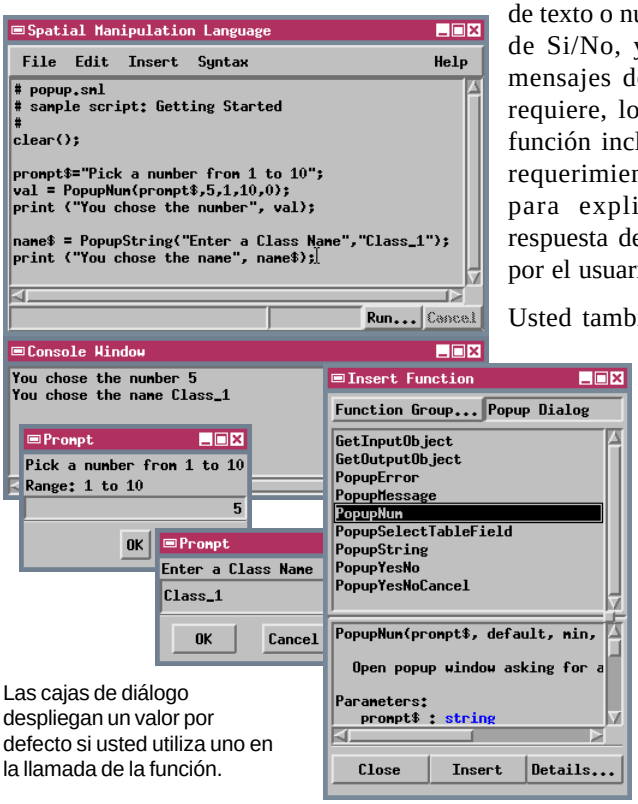
Ingreso del Usuario

PASOS
- ☑ despeje la ventana SML con File / New
- ☑ escriba en la ventana de la consola las declaraciones de prompt e input ilustradas y ejecute [Run] el guión
- ☑ escoja Insert / Function
- ☑ clic el botón Function Group, seleccione Popup Dialog de la ventana Function Group y clic [OK]
- ☑ escoja File / Open y seleccione LITEDATA / SML / POPUP.SML y [Run] el guión

El más simple tipo de ingreso y salida para el usuario utiliza la ventana de la consola. Usted puede imprimir mensajes de requerimiento y capturar respuestas del usuario usando las funciones print() e input$(). El código de ingreso en la consola es simple para el autor del guión, pero los mensajes en la consola podrían perderse para un usuario desatento.

```
clear(); print("Enter your name:")
name$ = input$()
print("Your name is: ",name$)
```

Las ventanas de diálogo desplegable ofrecen mayor flexibilidad y al mismo tiempo es menos probable que confundan al usuario. SML incluye funciones predefinidas en el grupo de funciones Popup Dialog que abren diálogos para el ingreso de valores de texto o numéricos, respuestas de Si/No, y el despliegue de mensajes de error. Cuando se requiere, los parámetros de la función incluye un mensaje de requerimiento que puede usar para explicar que valor o respuesta debería ser ingresado por el usuario.



Las cajas de diálogo despliegan un valor por defecto si usted utiliza uno en la llamada de la función.

Usted también puede construir sus propias ventanas de diálogo para proveer de una interface interactivo consistente para su guión. Estas ventanas pueden incluir botones, menús de listas y otros componentes con los que está familiarizado en la interface de TNTmips. Ejemplos mostrando como crear ventanas de diálogo con SML se incluyen más adelante en este folleto.

Bucles y Bifurcaciones

Bucles Implícitos. Cuando SML mira una variable de un objeto raster en el lado izquierdo de una declaración de asignación, ejecuta un bucle implícito, evaluando el lado derecho de la declaración y asignando el resultado a cada celda del objeto raster en el lado izquierdo:

```
R = R * scale  # multiplies each cell in R
```

Las declaraciones **For Each** para objetos rasters y vector tienen la forma:

```
for each Rastvar statement
for each Rastvar[lin,col] statement
for each Rastvar in Region statement
for each vector_element[n] in V statement
```

En la notación raster (R), `lin y col` indican el número de línea y columna de la "posición actual" en el raster para el acceso dentro del bucle de procesamiento. En la notación vector (V), los elementos vector pueden ser "point", "line", "poly", o "node". La n es opcional y puede ser omitida. Si se proporciona, la variable n se usa como el contador del bucle.

Las declaraciones **For** tienen dos formas:

```
for var=expr to expr statement
for var=expr to expr step expr statement
```

Bucles usando declaraciones "for" permiten un guión que opere en porciones de un conjunto de valores (celdas raster, valores de arreglos, números de elementos) especificados por rangos, o por "pasos" a través de un conjunto de valores.

While. Tenga cuidado de los bucles "while"

```
while (condition) statement
```

En tanto la condición del bucle retorne verdadero, el bucle continúa. Si la condición nunca llega a ser falsa, usted está ante un bucle infinito.

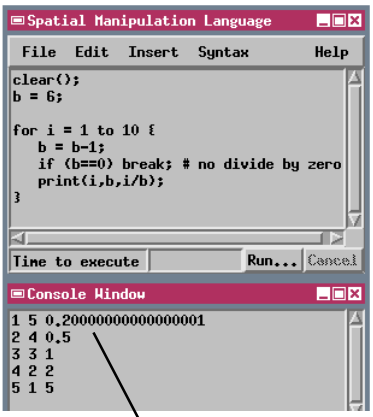
```
a = 0;
while (a <= 360) {
  print (a, sin(a/deg));
  a = a + 1;}
```

PASOS

- ☑ seleccione File / Open y seleccione WHILEFOR.SML de la carpeta LITEDATA / SML
- ☑ ejecute el guión
- ☑ cambie la condición while y ejecute el guión nuevamente
- ☑ cambie el valor del paso en el "for" del bucle y ejecute el guión nuevamente

NOTA: la secuencia de palabras clave "for each" también se puede escribir como una sola palabra: "foreach". Esta versión de SML no soporta comandos "foreach" anidados.

La declaración **break** se usa para salir de un bucle antes que éste pueda terminar de otra manera. Es a menudo usado en una prueba condicional dentro de un bucle. La declaración break previene en este ejemplo una división por cero.



Note que con todos los sistemas de computadoras, algunas operaciones dejan errores muy pequeños en valores de punto flotante (1/5 deja 0.20000000000000001).

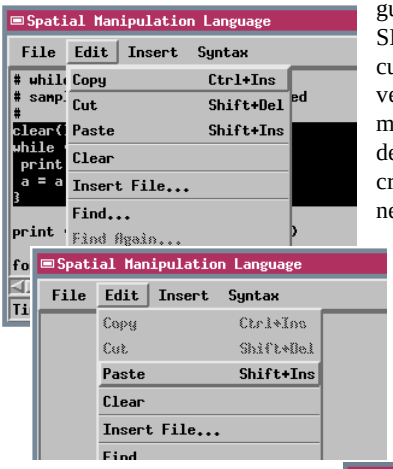
Desarrollo de Guiones y su Verificación

PASOS (no para Macintosh)

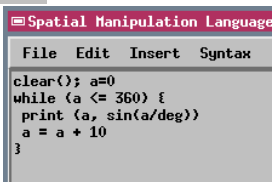
- ☑ mantenga el guión del ejercicio previo abierto
- ☑ abra otra instancia de la ventana SML con Process / SML / Edit Script
- ☑ mueva la nueva ventana SML de forma que no oculte a la primera
- ☑ seleccione el código del bucle "while" del guión WHILEFOR
- ☑ use las selecciones de Copy y Paste en los menúes Edit para copiar la sección seleccionada a su nuevo guión
- ☑ remueva el carácter de cierre "}" de su nuevo guión
- ☑ escoja Syntax / Check para el nuevo guión
- ☑ escoja File / Exit para la ventana del nuevo guión y no almacene los cambios

La forma más fácil de desarrollar un guión SML es adaptar a sus necesidades los ejemplos de guiones distribuidos con los productos TNT. Usted puede abrir dos ventanas de edición de guiones SML lado a lado y usar las funciones de menú "copiar – pegar", para copiar secciones de código desde el guión de ejemplo de MicroImages hacia el guión que está desarrollando. Si usted está bajo ejecutando las operaciones de Copia y Pegado bajo sistema operativo Windows, use el portapapeles de Windows, de forma que pueda también copiar y pegar textos entre el editor SML y un editor bajo Windows.

La opción de verificación en el menú Syntax chequea los errores de sintaxis de su guión. Los tipos de error que se pueden encontrar incluyen la falta de parámetros en una función, errores de escritura de funciones o variables, y paréntesis o bucles sin cerrar. El verificador de sintaxis no detecta errores de lógica tales como bucles infinitos o ingreso de valores incorrectos.



Si el verificador de sintaxis encuentra problemas en su guión, la línea de mensajes al final de la ventana SML, despliega un mensaje de error y coloca el cursor al final de la última parte del guión que el verificador pudo interpretar correctamente. A menudo el error se halla inmediatamente después de la ubicación del cursor, pero si el error involucra bucles de procesos anidados, usted podría necesitar buscar algunos pasos más adelante del cursor para encontrar el problema. Usted debería usar frecuentemente el verificador de sintaxis conforme va desarrollando el guión. Verifique cada pequeña porción que usted desarrolla. Es más fácil encontrar y arreglar los errores sobre la marcha en lugar de esperar para arreglar todos los errores en un guión largo y complejo. La verificación de la sintaxis también actualiza la lista de las variables disponibles para su ingreso desde la ventana Insert/Symbol.



NOTA: el MacOS no le permite abrir dos instancias del mismo proceso, de forma que usted no puede usar este método para copiar y pegar entre guiones SML en plataforma Mac.

Barra de Herramientas y el Menú Personalizado de SML

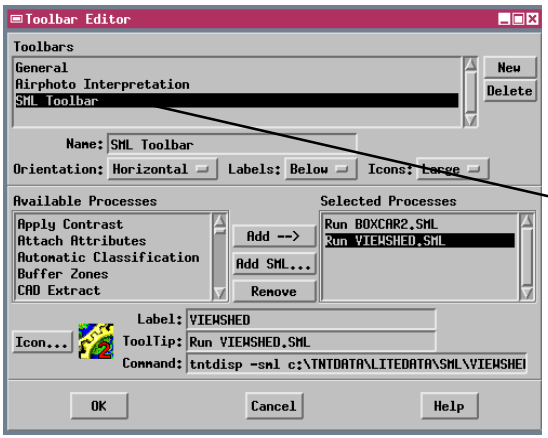
Usted puede seleccionar y ejecutar cualquier guión SML sin abrir la ventana de edición de SML, seleccionando SML /Run desde el menú Process. También puede asignar guiones SML a iconos en la barra de herramientas personalizada. Utilice la venta del Editor de la Barra de Herramientas para crear o seleccionar una barra de herramientas, fije la orientación a horizontal o vertical y defina las posiciones de las etiquetas. Luego seleccione uno o más guiones SML y edite las cajas de texto de las Etiquetas e información de las herramientas tal como se ilustra, para establecer el texto de la interface de cada uno. Presione el botón Icon para seleccionar un icono para cada guión. Los pasos en este ejercicio crean una nueva barra de herramientas SML con dos botones de iconos para los guiones.

Los ejemplos de guiones SML proporcionados por MicroImages pueden también ser ejecutados desde un menú desplegable personalizado si usted les ha instalado desde el CD-ROM de productos TNT. (Si su barra de menúes TNT no tiene el menú Custom, ejecute el programa de instalación desde el CD y seleccione **Install Sample SML Scripts**.) El programa de instalación crea un subdirectorio /CUSTOM y copia a este los ejemplos de guiones *.SML . De allí en adelante, cualquier nuevo guión que usted ponga en el directorio /CUSTOM estará disponible en el menú Custom.

PASOS

- ☑ escoja Toolbars / Edit en el menú principal TNTmips
- ☑ presione [New] en la ventana del Editor de Toolbar
- ☑ edite el campo Name para leer “SML Toolbar”
- ☑ seleccione Horizontal del menú Orientation
- ☑ clic [Add SML...]
- ☑ seleccione BOXCAR2.SML
- ☑ clic [Icon...] y seleccione un icono
- ☑ repita los dos pasos previos para VIEWSHED.SML
- ☑ clic [OK] para terminar

El menú desplegable Custom lista los guiones del subdirectorio /CUSTOM creados por el proceso de instalación de TNT



Utilice el Editor de la Barra de Herramientas para añadir los iconos de BOXCAR2 y VIEWSHED a una nueva barra de herramientas.

Objetos Guión y Encriptamiento

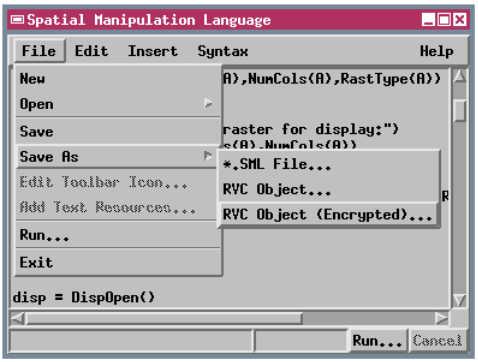
PASOS

- ☑ seleccione File / Open / *.SML File y escoja / LITEDATA / SML / BOXCAR2.SML
- ☑ seleccione File / Save As / RVC Object (Encriptado)
- ☑ cree un nuevo Archivo de Proyecto y objeto SML como lo requerido
- ☑ seleccione una palabra clave de encriptamiento en las opciones de la ventana Encryption
- ☑ si usted no esta utilizando TNTlite, use File / Open para seleccionar su guión encriptado (este solo muestra un mensaje de encripción)

NOTA: Se requiere de una llave física para ejecutar un objeto guión SML encriptado. Por consiguiente los guiones encriptados no pueden ejecutarse en TNTlite.

Hasta ahora usted ha trabajado con guiones SML que han sido almacenados como archivos de texto independientes con la extensión SML. Esto son archivos de texto de 1byte que pueden ser abiertos con cualquier editor de texto. Si usted edita un archivo de guión con otro editor de texto, asegúrese de almacenarlo con la extensión SML.

Un guión SML también se puede almacenar como un objeto guión en el Archivo de Proyecto (utilice File / Save As / RVC Object). Esto le permite colocar en el mismo archivo, los objetos de ingreso, salida y guiones si usted estima esto conveniente. Otra ventaja de almacenar un guión en un Archivo de Proyecto es la habilidad de **encriptar** el objeto guión. Usted podría distribuir sus guiones a otros pero aún proteger sus esfuerzos de desarrollo y algoritmos propios. Un objeto de guión encriptado puede ser ejecutado únicamente por usuarios autorizados de TNTmips y no puede ser mirado o editado por cualquiera (incluyendo el creador; siempre mantenga una copia sin encriptar del guión para referencia o desarrollos posteriores). Usted puede permitir que un guión encriptado sea ejecutado por cualquier usuario de TNTmips o limitar su uso a computadoras con un número específico de llave de la licencia de software. Usted también puede escoger el requerimiento de una palabra clave



Utilice la opción Save AS / Encrypted para crear una copia encriptada del guión en un Archivo de Proyecto. Si usted abre un guión encriptado en la ventana SML, este muestra solamente un mensaje de encripción. IMPORTANTE: **siempre mantenga una copia sin encriptar para la edición**

Objetos Raster

Un conjunto completo de funciones raster le permiten a sus guiones SML leer, crear y analizar objetos raster. Usted puede escribir expresiones matemáticas para calcular valores de nuevos objetos raster a partir de uno o más rasters de entrada o usar varias funciones SML de alto nivel para crear nuevos valores raster.

Utilice las funciones GetOutputRaster() y CreateRaster() para crear nuevos objetos raster. Cuando usted crea un objeto raster de salida, analice un poco las opciones sobre la especificidad del tipo de datos: binarios, enteros, con signo, sin signo, y punto flotante. Por ejemplo si los cálculos del guión SML pueden crear valores de salida de celdas negativas, asegúrese de especificar un tipo de datos con signo. Varias funciones proveen acceso a subobjetos raster.

El ejemplo de guión RATIOSCL está designado para calcular la razón entre dos bandas de una imagen raster (asumiendo que son rasters de 8bits sin signo) y rescalar el resultado al rango de datos de 8bits sin signo para el raster de salida. Los valores crudos resultantes de la razón podrían variara en el rango entre .004 (1/255) y 255, y un escalamiento separado es aplicado para razones menores que o mayores que 1. El factor de escala para el rango superior esta basado en el máximo valor de la razón para toda el área de la imagen. Esto necesita que se almacenen los valores crudos de la razón en un raster temporal de punto flotante, calcular el factor de escala a partir del valor de razón máximo, y luego calcular los valores rescalados y escribirlos al raster final de salida.

PASOS

- ☑ seleccione File / Open y seleccione RATIOSCL.SML de la carpeta LITEDATA / SML
- ☑ estudie la estructura del guión y la sintaxis de las declaraciones
- ☑ ejecute el guión
- ☑ cuando se requiera un raster para N, seleccione PHOTO_IR del Archivo de Proyecto CB_TM en LITEDATA / CB_DATA
- ☑ seleccione RED del Archivo de Proyecto CB_TM para ingresar el objeto D
- ☑ cree un nuevo objeto raster para RS
- ☑ para este ejercicio y los de las páginas siguientes, use el proceso de despliegue para mirar los objetos de ingreso y el nuevo objeto creado por el guión.

Raster de razón escalado (izquierda) producido por RATIOSCL.SML. a partir de CB_TM / RED (centro) y CB_TM / PHOTO_IR (derecha).

Objetos Vector

Una creciente lista de funciones soporta la creación, lectura, escritura y manipulación de objetos vector. Mire las definiciones de funciones vector en los grupos Vector, Vector Network y Vector Toolkit.

PASOS

- ☑ seleccione File / Open / *.SML File y abra el guión VECTCOMB.SML de LITEDATA / SML
- ☑ ejecute el guión utilizando para ingreso HYDROLOGY y ROADS desde CB_DATA / CB_DLG

```
CloseVector
CreateTempVector
CreateVector
FindClosestLabel
FindClosestLine
FindClosestNode
FindClosestPoint
FindClosestPoly
GetInputVector
GetInputVectorList
GetOutputVector
GetVectorLinePointList
GetVectorNodeLineList
GetVectorPolyAdjacentPolyList
GetVectorPolyIslandList
GetVectorPolyLineList
NumVectorLabels
NumVectorLines
NumVectorNodes
NumVectorPoints
NumVectorPolys
OpenInputVectorList
OpenVector
VectMerge
VectorAND
VectorElementInRegion
VectorExists
VectorExtract
VectorOR
VectorReplace
VectorSubtract
VectorToolkitInit
VectorXOR
```

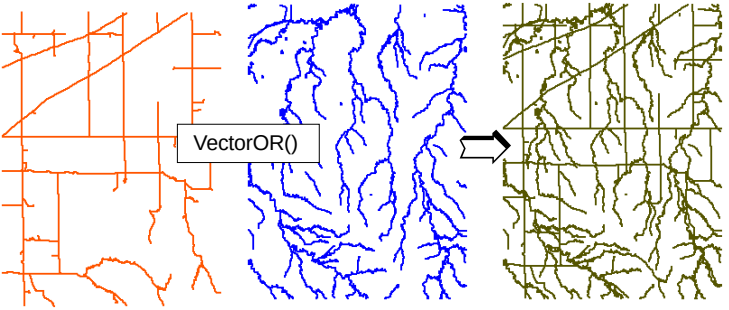
Las funciones Vector están listadas en las listas de funciones de Vector (arriba), Vector Network, y Vector Toolkit

Un guión simple ilustra las funciones básicas para el ingreso, salida y una de las combinaciones de vectores:

```
GetInputVector(Voperator);
GetInputVector(Vsource);
GetOutputVector(Vor);
Vor = VectorOR(Voperator, Vsource);
```

Las operaciones de extracción de vectores están soportadas por funciones similares. Por ejemplo, refiérase al guión de ejemplo CUSTOM / VECTOR / VECEXTR.SML.

SML también soporta más complejas interacciones entre objetos vector y objetos de otros tipos. Usted ya ha visto VIEWSHED.SML (página 4). Otro ejemplo se proporciona en CUSTOM / FOCAL / VECFOCAL.SML, la cual utiliza puntos en un objeto vector para seleccionar celdas en un objeto raster y aplica la función FocalMean() para cada una de esas celdas en su orden. Abra este guión y observe como las coordenadas del vector (x=V.point[i].Internal.x) son traducidas en coordenadas del mapa usando la función de georeferencia ObjectToMap(V,x,y,georefV,xVector,yVector), y como MapToObject(georefR, xVector, yVector, R, rCol, rLine) encuentra las celdas raster correspondientes a la coordenadas del mapa.



El guión corto mostrado arriba usa VectorOR() Para combinar dos objetos vector de ingreso en un solo objeto vector de salida.

Usando la Caja de Herramientas de Vector

Las funciones en el grupo Vector Toolkit le permiten a un guión modificar los elementos en un objeto vector existente o añadir nuevos elementos a un objeto. Para modificar un objeto vector existente, el guión debe primero inicializar la caja de herramientas de Vector para usar con ese objeto:

```
GetInputVector(V);
VectorToolkitInit(V);
     [Editing operations with vector
              toolkit functions]
CloseVector(V);
```

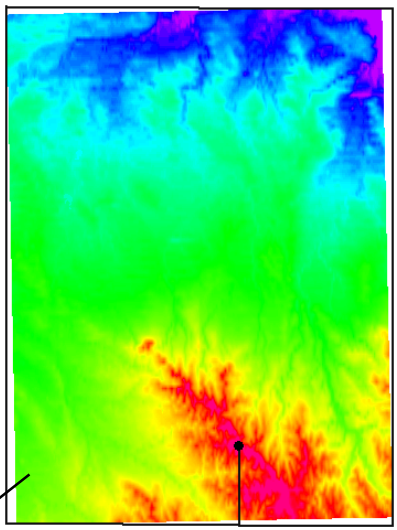
Cuando usted añadirá elementos a un nuevo objeto vector de salida, la inicialización de las herramientas puede hacerse cuando el objeto es creado. El segundo argumento de la función GetOutputVector() es una bandera string opcional, que puede ser usada para fijar el nivel de topología y para inicializar las herramientas vector. Por ejemplo fijando este argumento a "VectorToolkit,Polygonal" inicializa las herramientas vector y establece la topología poligonal para el objeto vector.

El guión de ejemplo VTOOLKIT.SML muestra como algunas de las funciones de las herramientas vector pueden ser usadas para crear elementos en un nuevo objeto vector. El guión primero abre un raster de ingreso y encuentra su extensión geográfica y la posición en el mapa de la celda con le mayor valor. El guión luego crea un nuevo objeto vector con una georeferencia implícita al objeto raster de ingreso, añade luego un elemento punto en la posición de la celda de máximo valor, y dibuja una línea vector por el perfil de la extensión del raster. Luego se encuentra la ubicación dentro de esta línea limítrofe que esté más cercana al punto de la celda de máximo valor y se añade una línea conectando estos dos puntos. Luego el objeto vector es validado (para verificar la topología y calcular los atributos estándar) y cerrado.

PASOS

- ☑ seleccione File / Open / *.SML File y abra el guión VTOOLKIT.SML de LITEDATA / SML
- ☑ estudie la estructura y comentarios del guión
- ☑ ejecute el guión usando para ingreso DEM_16BIT del Archivo de Proyecto CB_ELEV en LITEDATA / CB_DATA

```
ClosestPointOnLine
VectorAddLabel
VectorAddLine
VectorAddNode
VectorAddPoint
VectorAddTwoPointLine
VectorChangeLine
VectorChangePoint
VectorDeleteDangleLines
VectorDeleteLabel
VectorDeleteLabels
VectorDeleteLine
VectorDeleteLines
VectorDeleteNode
VectorDeleteNodes
VectorDeletePoint
VectorDeletePoints
VectorDeletePoly
VectorDeletePolys
VectorDeleteStdAttributes
VectorLineRayIntersection
VectorSetFlags
VectorSetZValue
VectorUpdateStdAttributes
VectorValidate
```



Raster DEM16_BIT y el objeto vector creado a partir de él por medio del guión de ejemplo.

Objetos CAD y TIN

Una creciente lista de funciones soportan la creación, lectura, escritura y manipulación de objetos CAD y TIN. El guión de ejemplo CAD.SML utiliza algunas de las numerosas funciones CAD. El guión utiliza un objeto raster como ingreso para definir la extensión geográfica y georeferenciación y crea un nuevo objeto CAD georeferenciado, al cual algunos elementos son añadidos. Un elemento círculo es dibujado centrado al punto geográfico central del raster, luego un elemento lineal se dibuja desde el centro a la circunferencia del círculo. Varios elementos tipo rectangular son añadidos alrededor del punto central.

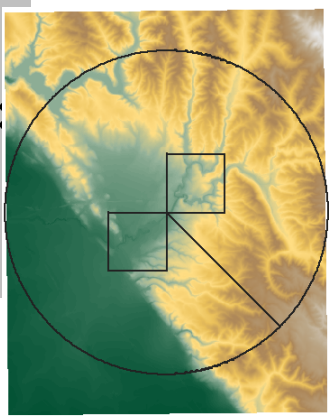
PASOS

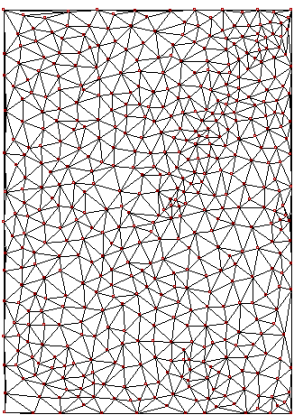
- ☑ seleccione File / Open / *.SML File y abra el guión CAD.SML de LITEDATA / SML
- ☑ examine y luego ejecute el guión usando el objeto raster HAYWARD del Archivo de Proyecto HAYWDEM en LITEDATA / SF_DATA
- ☑ abra el guión TIN.SML de LITEDATA / SML
- ☑ estudie y luego ejecute el guión, usando el objeto ELEV_PTS del Archivo de Proyecto SURFACE en LITEDATA / SURFMODL para el ingreso

El guión de ejemplo TIN.SML ilustra algunas de las funciones TIN. Utiliza la función TINCreateFromNodes() para hacer un nuevo objeto TIN a partir de un arreglo de coordenadas de nodos. Los arreglos de coordenadas son creados en este caso leyendo las coordenadas de los puntos de un objeto vector 3D. El guión también utiliza funciones para leer el número de marcos, bordes y triángulos.

```
CADAttachDBRecord
CADCreateBlock
CADElementInRegion
CADElementType
CADGetElementList
CADInsertBlock
CADNumBlocks
CADNumElements
CADReadArc
CADReadArcChord
CADReadArcWedge
CADReadBox
CADReadCircle
CADReadEllipse
CADReadEllipticalArc
CADReadEllipticalArcChord
CADReadEllipticalArcWedge
CADReadLine
CADReadPoint
CADReadPoly
CADReadText
CADUnattachDBRecord
CADWriteArc
CADWriteArcChord
CADWriteArcWedge
CADWriteBox
CADWriteCircle
CADWriteEllipse
CADWriteEllipticalArc
CADWriteEllipticalArcCh
CADWriteEllipticalArcWe
CADWriteLine
CADWritePoint
CADWritePoly
CADWriteText
CloseCAD
CreateCAD
GetInputCAD
GetOutputCAD
OpenCAD
```

```
CloseTIN
GetInputTIN
GetOutputTIN
TINAddNode
TINCreateFromNodes
TINDeleteEdgeAndMakeHole
TINDeleteNode
TINDeleteNodeAndMakeHole
TINDeleteTriangleAndMakeHole
TINDeleteTrianglesInPolygon
TINElementInRegion
TINGetConnectedEdgeList
TINGetConnectedNodeList
TINGetEdgeExtents
TINGetEdgeNodesAndTriangles
TINGetNodeExtents
TINGetNodeZValue
TINGetSurroundTriangleList
TINGetTriangleExtents
TINGetTriangleNodesAndEdges
TINGetTrianglesInPolygon
TINNumberEdges
TINNumberHulls
TINNumberNodes
TINNumberTriangles
TINSetNodeZValue
```





Objetos Región

Usted puede también crear y usar objetos región en los guiones SML. Los objetos Región representan el contorno de una región de interés en operaciones en otros objetos espaciales. Las funciones SML en el grupo de funciones Región le permite abrir y almacenar objetos región, verificar si coordenadas particulares de mapa se hallan dentro de una región, y ejecutar combinaciones de regiones (AND, OR, Subtract y XOR). Varias funciones en el grupo de Conversión de Objetos, le permiten convertir vectores y raster binarios en objetos región.

SML proporciona una manera simple de usar un objeto región para restringir acciones sobre un objeto raster: La construcción simple:

```
for each RastVar in RegionVar {
   [actions]
}
```

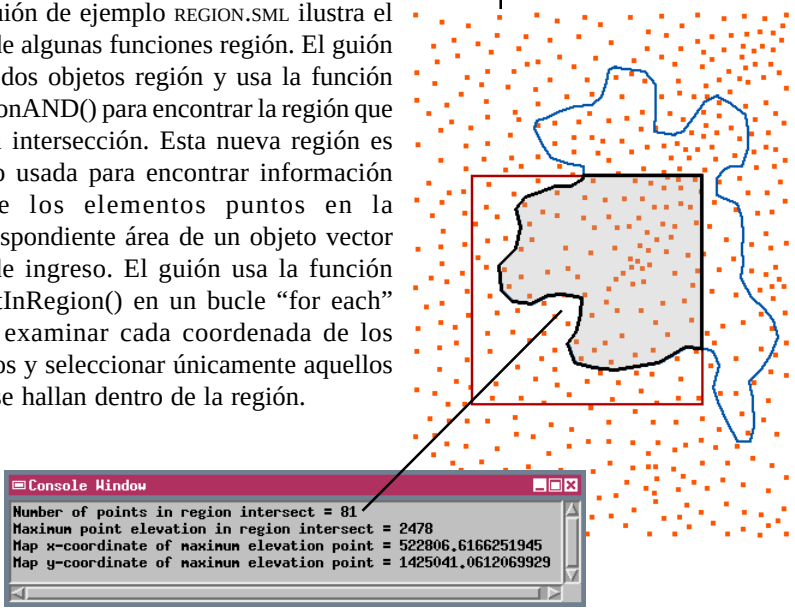
restringe las acciones a las celdas raster que se hallan dentro de los límites de la región. Esta construcción provee de una alternativa más simple que usando valores en una máscara raster binaria para controlar las operaciones.

El guión de ejemplo REGION.SML ilustra el uso de algunas funciones región. El guión abre dos objetos región y usa la función RegionAND() para encontrar la región que es su intersección. Esta nueva región es luego usada para encontrar información sobre los elementos puntos en la correspondiente área de un objeto vector 3D de ingreso. El guión usa la función PointInRegion() en un bucle “for each” para examinar cada coordenada de los puntos y seleccionar únicamente aquellos que se hallan dentro de la región.

PASOS

- ☑ seleccione File / Open / *.SML File y abra el guión REGION.SML de LITEDATA / SML
- ☑ estudie la estructura y comentarios del guión
- ☑ ejecute el guión usando para ingreso los objetos región POLYREGION y RECTANGLE del Archivo de Proyecto REGION en LITEDATA / SML y el objeto vector ELEV_PTS del Archivo de Proyecto SURFACE en LITEDATA / SURFMODL.

```
ClearRegion
CopyRegion
CreateRegion
GetInputRegion
GetOutputRegion
OpenRegion
PointInRegion
RegionAND
RegionOR
RegionSubtract
RegionTrans
RegionXOR
SaveRegion
```



Objetos Base de Datos

PASOS

- ☑ abra el guión de ejemplo DATABASE.SML desde la carpeta LITEDATA / SML
- ☑ ejecute el guión usando para ingreso el objeto HSOILS del Archivo de Proyecto HAYWSOIL en la carpeta LITEDATA / SF_DATA
- ☑ abra el guión de ejemplo DB2.SML de la carpeta LITEDATA / SML
- ☑ ejecute el guión usando para ingreso el objeto CB_SOILSLITE del Archivo de Proyecto CB_SOILS en la carpeta CB_DATA

El guión de ejemplo DATABASE.SML muestra como leer valores de atributos desde una base de datos. La sintaxis es una extensión de la construcción TABLENAME.FIELDNAME usada en las consultas. En un guión SML, la referencia a un campo de la base de datos debe también especificar al objeto, el subobjeto de base de datos (una base de datos separada se mantiene para cada tipo de elemento en un objeto vector o TIN), y el número de elemento. Si el campo siendo leído es un campo textual, usted debe también añadir el carácter "$" al final del campo de referencia:

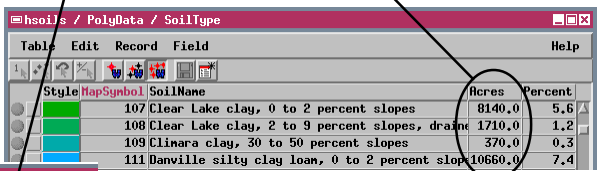
string$ = Vect.poly[4].table.field$.

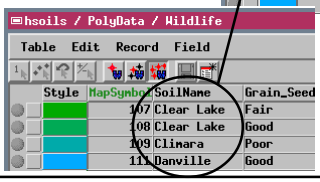
Funciones para crear y modificar bases de datos se hallan en el grupo de funciones Database. Este grupo incluye funciones para crear nuevas tablas, para añadir o insertar campos en las tablas, para escribir nuevos registros en la tabla, y para enlazar registros a elementos en el objeto espacial. El guión de ejemplo DB2.SML provee de ejemplos de estas operaciones. Este crea un nuevo objeto vector con puntos localizados en los centroides de los polígonos en el objeto vector de ingreso, crea una base de datos de puntos y una tabla, y copia los atributos seleccionados desde cada polígono al elemento punto asociado.

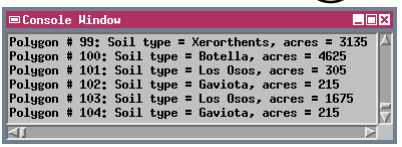
```
DatabaseGetTableInfo
FieldGetInfoByName
FieldGetInfoByNumber
NumRecords
OpenCADDatabase
OpenDatabase
OpenRasterDatabase
OpenTINDatabase
OpenVectorLineDatabase
OpenVectorPointDatabase
OpenVectorPolyDatabase
RecordDelete
TableAddField
TableAddFieldFloat
TableAddFieldInteger
TableAddFieldString
TableCopyToDBASE
TableCreate
TableExists
TableGetInfo
TableInsertFieldFloat
TableInsertFieldInteger
TableInsertFieldString
TableKeyFieldLookup
TableLinkDBASE
TableNewRecord
TableOpen
TableReadAttachment
TableReadFieldNum
TableReadFieldStr
TableWriteAttachment
TableWriteRecord
```

```
# Database.sml
# sample script: Getting Started
#
PopupMessage("Select /litedata/sf_data/HAYWSOIL/hsoils");
GetInputVector(V);
numpolys = NumVectorPolys(V);

for i = 1 to numpolys{
   acres = V.poly[i].SoilType.Acres;

   type$ = V.poly[i].Wildlife.SoilName$;

   printf("Polygon # %d: Soil type = %s, acres = %d\n",i,type$,acres);
}
```

DATABASE.SML refiere al campo ACRES de la tabla SOILTYPE y al campo SOILNAME de la tabla WILDLIFE.







Convirtiendo Objetos

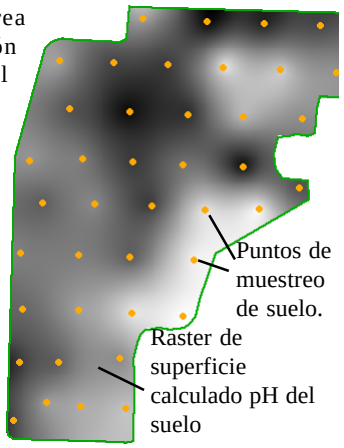
Una razón común para crear guiones SML es el deseo de automatizar secuencias de procesamiento de múltiples pasos, que requieren ser ejecutadas repetidamente en una cantidad de diferentes conjuntos de datos de ingreso. La habilidad de convertir datos geoespaciales de un tipo a otro dentro de SML, le proporciona una gran flexibilidad para diseñar tales guiones. El proceso estándar de conversión de datos en TNTmips lidera la industria en soporte de tipos de datos y funcionalidad. Muchos de estos procesos de conversión están disponibles como funciones SML en el grupo de funciones de Conversión de Objetos. Otras funciones especializadas de conversión en el grupo de funciones Surface Fitting interpolan una superficie raster partir de un objeto de ingreso vector o TIN.

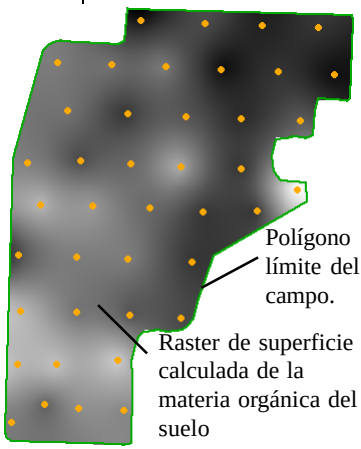
El guión de ejemplo SOILTEST.SML automatiza el procesamiento de datos de muestras de suelo y usa varios tipos de funciones de conversión de objetos. El guión lee una serie de valores químicos almacenados en una tabla de una base de datos enlazada a los elementos puntos vector de ingreso que representan las ubicaciones de las muestras. Para cada tipo de valor (pH del suelo, contenido de materia orgánica, y otros), el guión usa una función de ajuste de superficie para crear un raster de superficie. En pasos intermedios el guión usa un polígono vector que representa el borde del campo para crear un raster en blanco para usarlo como una máscara para cada superficie. También crea una región desde el polígono y la utiliza para escribir un valor de 1 en cada celda de la máscara raster que queda dentro del límite del campo.

PASOS

- ☑ abrir el guión de ejemplo SOILTEST.SML de LITEDATA / SML
- ☑ estudiar el guión, luego ejecutarlo usando los objetos en el Archivo de Proyecto SOILTEST en LITEDATA / SML para ingreso. Use el objeto SAMPPTS para los "Points" y el objeto BOUNDARY para "Boundary"
- ☑ acepte los valores por defecto para los otros parámetros requeridos por las ventanas de diálogos desplegables.

```
BinaryRasterToRegion
ConvertCMYKtoRGB
ConvertHBStoRGB
ConvertHIStoRGB
ConvertHSVtoRGB
ConvertRegionToVect
ConvertRGBtoHBS
ConvertRGBtoHIS
ConvertRGBtoHSV
ConvertVectorPolysToRegion
ConvertVectorPolyToRegion
ConvertVectToRegion
RasterCompositeToRGB
RasterRGBToComposite
RasterToCADBound
RasterToCADLine
RasterToTINIterative
RasterToVectorBound
RasterToVectorContour
RasterToVectorLine
TINToRaster
TINToVectorContour
VectorElementToRaster
VectorToBufferZone
```





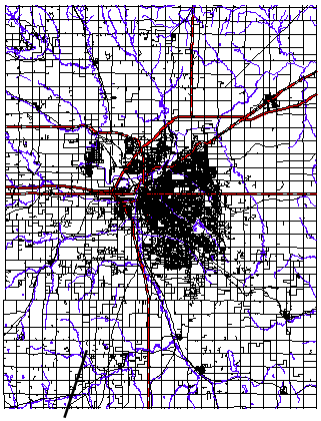
Ejemplo de Guión: Extraer Polígonos

PASOS

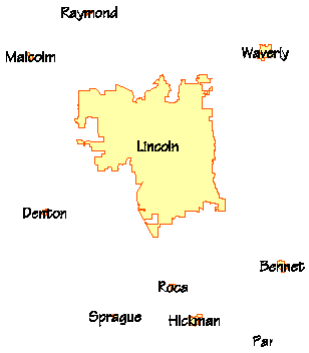
- ☑ escoja File / Open / *.SML File y seleccione de su directorio principal TNT CUSTOM / VECTOR / TIGER.SML
- ☑ estudie la estructura y comentarios del guión

El guión de ejemplo TIGER.SML proporciona un ejemplo de procesamiento en SML de vector y base de datos. Este extrae líneas especificadas de los objetos vectores de ingreso y los escribe en un objeto vector de salida, transfiere los atributos de la línea a los atributos de los polígonos de salida.

TIGER.SML fue diseñada para procesar objetos vector importados de archivos de líneas TIGER (versión 2000) producidos por la Oficina de Censos de los Estados Unidos. Los geodatos TIGER están organizados por condado, e integran líneas de geodatos de muchos tipos (hidrología, vías, líneas limítrofes de censos y administrativas) en un solo nivel de datos vector. Los polígonos topológicos resultado de la intersección de estos varios tipos de líneas con excepción de los polígonos individuales tienen poco significado geográfico. Los atributos de área están codificados únicamente como atributos de los lados derecho e izquierdo de las líneas. Esta característica de los datos TIGER hace difícil de acceder y desplegar información de área usando los objetos vector crudos.



Vector TIGER para un condado con estilos de líneas basados en sus atributos.



Polígonos de ciudad con etiquetas, extraídos para el mismo condado

Las líneas de límite de área en el vector TIGER, tales como límites de ciudades y pueblos, pueden ser identificadas por la desigualdad de los valores de atributos particulares en cada lado de la línea. Este guión encuentra las líneas límites de la ciudad en uno o más objetos vector TIGER de ingreso y escribe cada línea a un nuevo objeto vector de salida. Cuando todos los elementos línea de un particular límite de ciudad han sido transferidos, ellos se interceptan para formar un polígono en el vector de salida. Si la línea actual completa un nuevo polígono, el nombre de ciudad se lee de la base de datos de la línea de ingreso, y se crea un nuevo registro de la base de datos de polígonos conteniendo el nombre para el vector resultante. Este guión ha sido usado en MicroImages para procesar todos los 93 objetos vector TIGER de condados para el estado de Nebraska para producir un objeto de polígonos únicos de ciudades a nivel del estado.

Mas acerca del guión de extracción de polígonos se halla disponible en un documento en línea en:

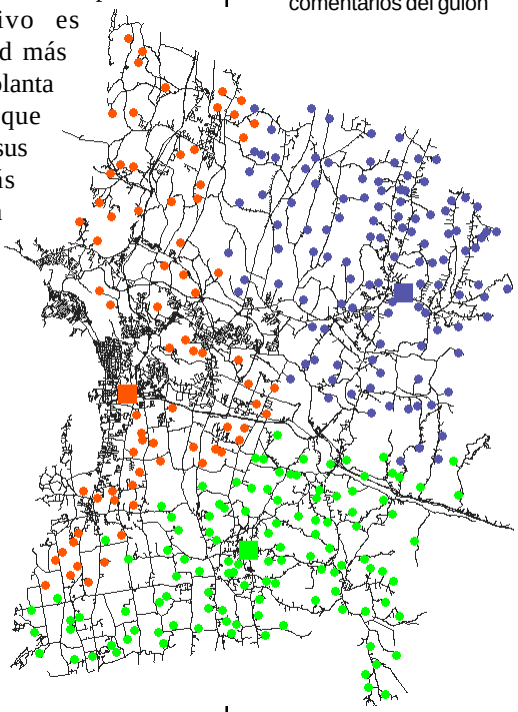
http://www.microimages.com/relnotes/v65/smltiger.pdf

Ejemplo de Guión: Red de Ruteo

El ejemplo de guión NETWORK.SML muestra una aplicación más complicada de procesamiento SML de vectores y base de datos. Esta utiliza las funciones de análisis de redes para afrontar el problema de una eficiente entrega de materiales desde locaciones numerosas y dispersas (tal como granjas) a un pequeño número de destinos (tal como plantas de procesamiento). El objetivo es determinar la distancia de red más corta desde cada granja a cada planta de procesamiento, de forma que cada granja pueda transportar sus productos a la planta más cercana. Se requiere de un guión para resolver este problema considerando que las granjas y las plantas son representadas como puntos en objetos vector distintos, separados del objeto que contiene la red de caminos.

PASOS

- ☑ escoja File / Open / *.SML File y seleccione de su directorio principal de TNT CUSTOM / VECTOR / NETWORK.SML
- ☑ estudie la estructura y comentarios del guión



Ejemplos del resultado del guión de redes. Las ubicaciones de las granjas (círculos) han sido puestas con un estilo en el mismo color que las plantas de procesamiento (cuadrados) que se hallan más cercanas a lo largo de la red de caminos.

Para cada granja y planta de procesamiento, el guión añade un nodo al objeto de caminos en el punto más cercano de la línea más cercana. Se mantiene un control del número de elementos de estos dos conjuntos de nodos añadidos en un par de arreglos de forma que las distancias de red pueden ser asociados con la planta y granja correctas. Las funciones de análisis de redes se utilizan luego para calcular el conjunto de distancias requeridas, las cuales son almacenadas en una nueva tabla de la base de datos para el vector representando las granjas. Para cada punto de una granja, existe un registro enlazado por cada planta de procesamiento, mostrando la distancia de red mínima.

Más acerca del guión de redes se halla disponible en un documento en línea en:

http://www.microimages.com/relnotes/v65/smlnz.pdf

Incluyendo Guiones y Programas Ejecutables

PASOS

☑ desplace hasta el final la lista en la ventana Insert Operator para mirar las directivas de preproceso SML

El proceso SML incluye un conjunto de directivas de procesamiento previo que son interpretadas antes de todas las declaraciones regulares del guión. Las directivas de preproceso le permiten llamar a otros guiones y fijar modos alternativos del guión.

Usted puede hacer que un guión lea y ejecute otro guión SML utilizando la directiva $include:

```
$include "another.sml"
```

El guión incluido debe hallarse en el mismo directorio o Archivo de Proyecto que el guión padre. Si usted tiene varios guiones que necesitan usar las mismas funciones definidas por el usuario, la definición de funciones pueden hacerse en un guión separado que usted "$include" en los otros guiones.

Las directivas de preprocesamiento SML que se pueden insertar mediante la ventana Insert Operator son:

```
$ifdef
$ifndef
$else
$endif
$define
$include
```

Mientras usted está desarrollando un guión complejo podría desear disponer de un modo "normal" de ejecución y de un modo "de depuración" que imprima a la consola la información relevante para ayudarle a identificar posibles puntos de falla. Usted puede fijar el modo de depuración usando la directiva:

```
$define DEBUG
```

y coloque entre paréntesis todas las declaraciones de depuración con el siguiente par de directivas:

```
$ifdef DEBUG
    [serie de declaraciones de impresión]
$endif
```

Para ejecutar el guión en el modo normal, simplemente comenta la única declaración $define, dejando su código de depuración sin cambios para un uso futuro.

Cuando utiliza la función run() como se muestra a la derecha, SML espera hasta que usted cierre el programa externo antes de continuar con la siguiente declaración en el guión. Si usted fija el valor del argumento opcional "wait" de la función run() a 0, el programa externo se ejecuta en un segundo plano.

Si su guión requiere manipulaciones y conversiones que no se hallan soportadas por SML usted puede usar la función del sistema run(), para llamar a procesos TNT o programa externos. Por ejemplo, la versión actual de SML no incluye una función para convertir un objeto CAD a un objeto Vector, usted podría seleccionar que su guión ejecute Prepare / Convert / CAD to Vector:

Para encontrar el nombre de un módulo de procesos de TNT (tal como "convcobjs cadtovec" en el ejemplo de la derecha), utilice cualquier editor de texto para abrir el archivo TNTMIPS.MNU (el que se halla en su directorio de TNT)

run ("c:/tnt/win32/convobjs cadtovec").

Un Nivel SML en el Despliegue

El botón del icono Add SML

El proceso estándar de despliegue (Display / Spatial Data) soporta el uso de un guión SML como un nivel más, de igual forma que lo puede ser son un objeto raster, vector, CAD, o TIN. Un nivel de guión SML puede usar funciones de dibujo cartográfico flexibles, para crear símbolos especiales de mapas y recuadros.

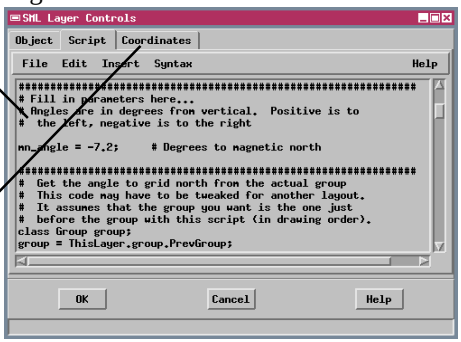
El guión de ejemplo ARROW.SML está diseñado para dibujar en el formato un símbolo de mapeo de la orientación de la declinación magnética. El nivel SML debe ir solo en un grupo. Este determina la dirección del norte verdadero a partir del mapa en el grupo previo del formato. El guión de ejemplo neatline.msl dibuja un recuadro alrededor de un grupo, e incluye ítems de dibujo adicionales que usted puede encender removiendo el carácter de comentario (#) de las declaraciones relevantes del guión.

PASOS

- ☑ ejecute Display / Spatial Data y abra un grupo New 2D
- ☑ clic Add SML
- ☑ seleccione la tarjeta Script en la ventana SML Layer Controls y escoja File / Open / *.SML
- ☑ seleccione LITEDATA / SML / ARROW.SML
- ☑ en el panel Coordinates, use el botón Projection para cambiar el sistema de coordenadas a Universal Transverse Mercator
- ☑ clic [OK] para cerrar la ventana Layer Controls
- ☑ examine el despliegue, luego remueva el nivel SML
- ☑ añada el objeto _8_BIT del Archivo de proyecto CB_COMP en LITEDATA / CB_DATA
- ☑ clic Add SML y seleccione LITEDATA / SML / NEATLINE.SML
- ☑ en el panel Coordinates, fije el sistema de coordenadas a United States State Plane 1927 y la Zona to Nebraska North
- ☑ clic [OK] para cerrar la ventana Layer Controls

El panel de tarjeta Script en la ventana SML Layer Control contiene la interface para editar y ejecutar guiones.

El panel Coordinates le permite relacionar su nivel de guión a las coordenadas del mapa de los otros niveles en el despliegue.



SML y GeoFórmulas

Un Folleto Tutorial separado está dedicado al tópico de GeoFórmulas. Mirar el *Tutorial: Usando GeoFórmulas"*.

PASOS

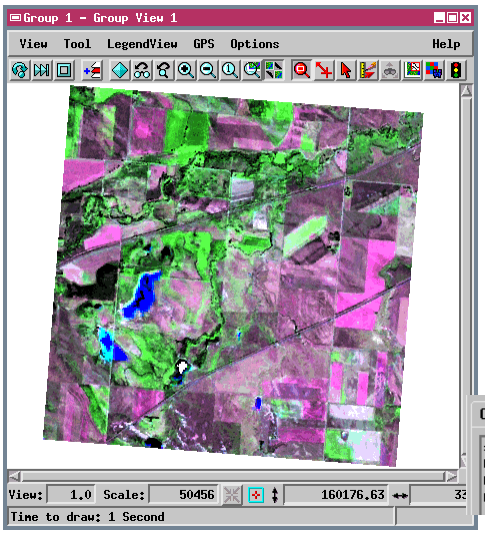
- ☑ escoja Display / Spatial Data
- ☑ clic Add Geoformula / Quick-Add Geoformula 
- ☑ seleccione LITEDATA / GEOFRMLA / BROV_UMN.GSF
- ☑ para ingreso, seleccione tres bandas TM del Archivo de Proyecto CB_TM y la imagen SPOT_PAN en el Archivo de Proyecto CB_SPOT, todos los objetos se hallan en LITEDATA / CB_DATA

El ejemplo LITEDATA / GEOFRMLA / BROV_UMN.GSF ilustra el realce dinámico de una imagen de baja resolución TM con una SPOT de alta resolución.

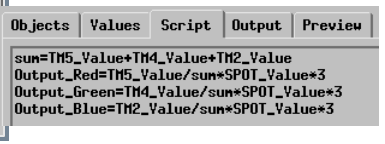
Un nivel de GeoFórmula es un nivel de despliegue calculado, que utiliza uno o más objetos de entrada para derivar un resultado a desplegar. Esto le proporciona una forma de aplicar operaciones SML a objetos en forma "instantánea" en lugar de ejecutar procesos separados para preparar objetos resultantes para su despliegue. Un nivel de GeoFórmula contiene un "objeto virtual"; éste no crea un objeto de salida que sea almacenado en un Archivo de Proyecto. En cambio crea un nivel de despliegue que libera todos sus recursos del sistema (tales como espacio de disco y memoria) cuando termina con él.

Por ejemplo, las bandas roja e infrarroja de una imagen raster pueden ser combinadas para producir un Índice Transformado de Vegetación (TVI). Por supuesto que TNTmips ofrece un proceso simple que produce un raster de salida con el TVI a partir de los objetos de ingreso seleccionados, si usted desea mantener el resultado del TVI para otros usos. Pero si únicamente desea mirar el resultado del TVI y no le preocupa mantener el objeto resultante, usted puede usar un nivel de despliegue de GeoFórmula.

Un guión de GeoFórmula puede almacenarse como un archivo reutilizable. Un nivel de GeoFórmula puede combinarse con cualquier número de otros niveles en el proceso de despliegue de TNT, para crear una visualización compleja de múltiples objetos geoespaciales.



La característica de GeoFórmula es primariamente provista para tarea de visualización dinámica en el proceso de despliegue. Usted también puede ejecutar un proceso separado de GeoFórmula (Interpret / Raster / Combine / GeoFormula) para crear objetos permanentes de salida para otros usos.



Creando una Ventana de Diálogo Simple

Para guiones complejos que incluyen múltiples operaciones incluyendo ingresos del usuario, considere la creación de ventanas de diálogo personalizadas, para controlar la interacción del usuario con el guión. Las funciones SML en el grupo de funciones Widget (ventanas de acceso) proporciona acceso al conjunto de ventanas de acceso de Motif, el cual es utilizado para crear todas las ventanas en la versión X Window de los productos TNT.

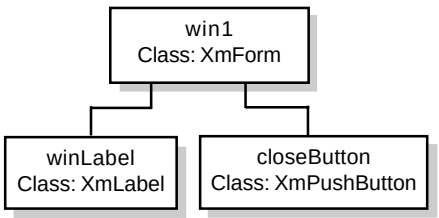
PASOS

- ☑ seleccione Process / SML / Edit Script
- ☑ escoja File / Open / *.SML File y seleccione DIALOG1.SML de la carpeta LITEDATA / SML
- ☑ ejecute el guión
- ☑ estudie las secciones del guión que definen las diferentes partes de la ventana de diálogo Hello World
- ☑ presione [Close] en la ventana Hello World

Una ventana de diálogo consiste de una ventana de acceso (widget) padre que contiene otros componentes de ventanas de acceso (widgets). Cada tipo de ventana de acceso (widget) es una clase separada en X y en SML. El guión de ejemplo DIALOG1.SML crea y abre una muy simple ventana de diálogo, que despliega una etiqueta de texto y tiene un botón de "Cierre". Cada uno de estos componentes es una ventana de acceso separada contenida en una widget XmForm. Una widget XmForm le permite colocar sus "hijos" (widgets contenidos) un esquema simple de ubicación relativa. Cada widget puede ser enlazado a otro en el tope, fondo, izquierda, o derecha, y usted puede especificar el valor de separación (en pixeles de pantalla) para cada lado. En este ejemplo el widget de etiqueta (class XmLabel) esta enlazado al formulario a los lados superior, izquierdo y derecho. El Botón de Cierre (class XmPushButton) está enlazado en su borde superior al widget de etiqueta y a los lados izquierdo y derecho al formulario. El formulario del widget automáticamente cambia de tamaño para acomodar todos los widget contenidos.

```
# DIALOG1.SML
# Sample script for Getting Started.
# Creates and opens a simple dialog window.

# Define parent widget for dialog window.
class XmForm win1;

# Procedure for closing window
proc OnClose() {
   DialogClose(win1);
   DestroyWidget(win1);
}

# Set up dialog window
win1 = CreateFormDialog("Hello World");
win1.MarginHeight = 5;
win1.MarginWidth = 5;

# Create label text for window
class XmLabel winLabel;
winLabel = CreateLabel(win1,"Sample Dialog Window");
winLabel.TopWidget = win1;
winLabel.LeftWidget = win1;
winLabel.LeftOffset = 10;
winLabel.RightWidget = win1;
winLabel.RightOffset = 10;

# Create Close button attached to label on
# on top and to window margin on left and right
class XmPushButton closeButton;
closeButton = CreatePushButton(win1,"Close");
closeButton.TopWidget = winLabel;
closeButton.TopOffset = 5;
closeButton.leftWidget = win1;
closeButton.rightWidget = win1;
closeButton.bottomWidget = win1;
WidgetAddCallback(closeButton.ActivateCallback,OnClose);

# Open dialog window and keep script active
# until window is closed.
DialogOpen(win1);
DialogWaitForClose(win1);
```

Jerarquía del Widget en la ventana Hello World



Usted puede crear una ventana deslizable usando el widget contenedor XmScrolledWindow en lugar de XmForm, u organizar widgets hijos en una grilla usando el widget contenedor XmRowColumn.

Usando Widgets para Construir Ventanas de Diálogo

PASOS

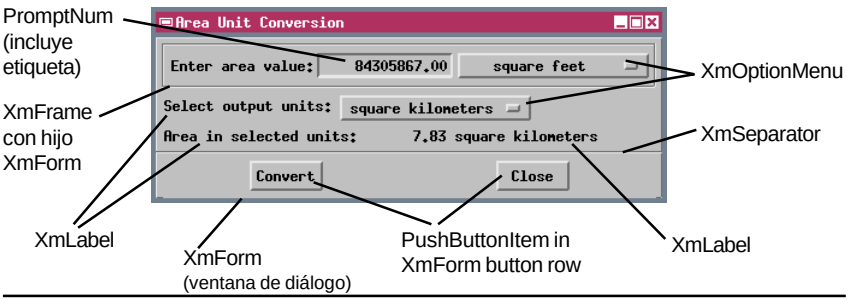
- ☑ escoja File / Open / *.SML File y seleccione DIALOG2.SML de la carpeta LITEDATA / SML
- ☑ ejecute el guión
- ☑ en la ventana de diálogo abierta por el guión, ingrese un valor en el campo Enter Area Value
- ☑ escoja una unidad para el área de ingreso en el menú superior de unidades
- ☑ escoja una unidad de área para la salida del menú de unidades inferior
- ☑ Presione el botón Convert
- ☑ estudie las secciones del guión que definen los diferentes componentes y acciones de la ventana
- ☑ clic [Close] cuando termine de trabajar con la ventana de diálogo

NOTA: El guión DIALOG2.SML fue escrito para usar una amplia variedad de tipos de widgets, y no para proveer un ejemplo de un buen diseño de ventanas o procesamiento eficiente. Un más eficiente diseño omitiría el botón Convert y recalcularía el valor de salida cuando cualquier de los parámetros del usuario sean cambiados.

El rol del botón de Cierre en el guión DIALOG1 está definido por el registro de un *callback* en el widget, usando la función WidgetAddCallback(). Un callback sirve como un puntero hacia una función o procedimiento que un widget llama en respuesta a uno o más eventos. Un XmPushButton tiene disponible un miembro de clase ActivateCallback para registrar la llamada a ser activada cuando el botón es presionado. En este ejemplo, activando el botón de cierre, se llama al procedimiento OnClose, definido al inicio del guión.

El guión de ejemplo DIALOG2.SML crea una ventana de diálogo más compleja que utiliza una variedad adicional de tipos de widget, incluyendo una campo para el ingreso de valores numéricos, un recuadro, una línea de separación y dos menúes de opciones. Los botones al final de la ventana utilizan diferentes clases de widgets que los botones del guión anterior. Estos son instancias de la clase PushButtonItem, los cuales pueden ser utilizados bien sea para botones de texto o botones de iconos. Los botones de texto deben ser ubicados en una fila de botones, y un tipo específico de XmForm y botones de iconos se deben colocar en un widget XmRowColumn. Usted no necesita usar la función WidgetAddCallback() para definir la acción de un PushButtonItem; la función que define el ítem, requiere el nombre de la función callback o procedimiento como uno de sus argumentos. El widget de menú de opción de unidad, también usa el último método para definir el procedimiento llamado cuando la unidad es cambiada.

Clases Widget usadas para crear la ventana de diálogo Area Unit Conversion



Creando y Usando un Área de Dibujo

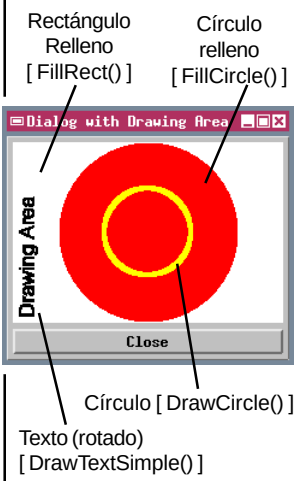
En algunas instancias usted podría desear diseñar una ventana de diálogo que incorpore un gráfico creado mediante el ingreso de sus datos o desde la salida de un proceso, o algún otro gráfico. Numerosas funciones en el grupo de funciones de Dibujo le permiten dibujar líneas, formas geométricas y textos, así como fijar el color y otras características de estilo. Para utilizar estas funciones usted debe incluir un widget XmDrawingArea en su ventana de diálogo.

El guión DIALOG3.SML ilustra como estructurar y utilizar un área de dibujo en una ventana de diálogo. Cuando usted crea el área de dibujo, se especifica su alto y ancho en pixeles de pantalla conjuntamente con el widget principal y los ambientes de enlace. La ubicación de los elementos en el área de dibujo están referidos a un sistema de coordenadas X-Y con unidades de pixeles de pantalla y origen (posición 0,0) en la esquina superior izquierda del área de dibujo. Cuando utiliza funciones como SetColor(), SetLineWidth(), y DrawTextSetFont(), estos parámetros se usan por las funciones de dibujo subsecuentes hasta que usted invoca nuevamente la función "Set" para cambiar los parámetros. Estos parámetros son almacenados en una estructura llamada contexto gráfico, el cual es creado por la función CreateGCForDrawingArea(). El GC debe también ser activado por la función ActivateGC() antes de que pueda ser utilizado.

Si su ventana de diálogo es cubierta por otra ventana y luego descubierta, los widgets regulares Xm son redibujados automáticamente. Sin embargo si utiliza un área de dibujo, su guión debe en forma explícita manejar este evento. Usted debe añadir a la lista de llamadas del widget del área de dibujo un ExposeCallback. Esta llamada es activada automáticamente cuando se abre la ventana o expuesta de otra manera. Todas las instrucciones del dibujo deben ser ubicadas dentro del proceso de llamada, de forma que el dibujo es activado por cualquier evento de exposición. Un contexto gráfico requiere una ventana activa, de forma que el GC debe también ser creado y activado dentro de la llamada.

PASOS

- ☑ escoja File / Open / *SML File y seleccione DIALOG3.SML de la carpeta LITEDATA / SML
- ☑ ejecute el guión
- ☑ estudie las secciones del guión que definen los diferentes componentes y acciones
- ☑ clic [Close] cuando haya terminado de trabajar con la ventana de diálogo



El guión DIALOG3.SML usa un widget de área de dibujo para dibujar (en orden) un rectángulo relleno en blanco, un círculo rojo relleno, un círculo amarillo, y un texto simple. El guión para la Herramienta de Perfil Raster, descrito en una página posterior, incluye un ejemplo más complejo del uso de un área de dibujo.

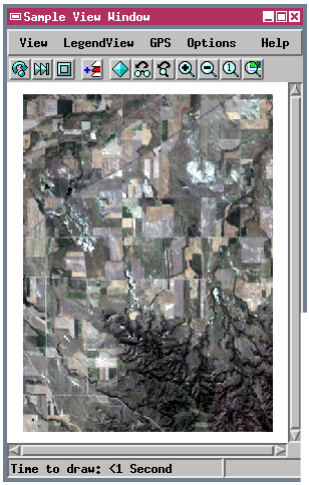
Creando un Despliegue en una Ventana de Diálogo

Una ventana de diálogo creada por un guión SML puede desplegar objetos de ingreso y salida en una ventana de despliegue. La función GroupCreateView() es utilizada para crear un widget de despliegue, para desplegar un grupo de geodatos dentro del diálogo principal. Otras funciones en los grupos de funciones Geodata Display, Geodata Display Group, Geodata Display Layout, y Geodata Display View, le permiten estructurar un grupo para despliegue, para añadir objetos y para acceder a información de escala y coordenadas.

El guión de ejemplo VIEW.SML muestra los pasos básicos requeridos para abrir una ventana de despliegue de un grupo y desplegar un raster de ingreso. El guión de ejemplo BOXCAR2.SML crea una más compleja ventana de diálogo, incorporando una cantidad de otros widgets adicionalmente al de despliegue.

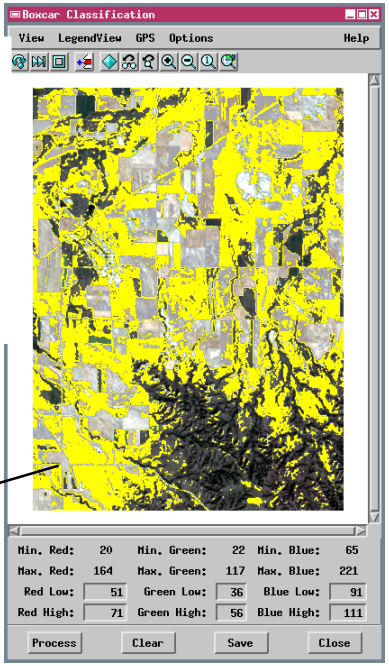
PASOS:

- ☑ seleccione File / Open / *.SML File y abra LITEDATA / SML / VIEW.SML
- ☑ Ejecute el guión usando como raster de ingreso _8_BIT desde el Archivo de Proyecto CB_COMP en LITEDATA / CB_DATA
- ☑ seleccione View / Close para cerrar la ventana



- ☑ seleccione File / Open / *.SML File y abra LITEDATA / SML / BOXCAR2.SML
- ☑ ejecute el guión, seleccionando como rasters de ingreso RED, GREEN, y BLUE desde el Archivo de Proyecto CB_TM en LITEDATA / CB_DATA
- ☑ presione [Process] en la ventana Boxcar Classification para ejecutar usando los valores por defecto
- ☑ estudie el guión para mirar como los distintos componentes de la ventana son estructurados y como las acciones son controladas

El guión de ejemplo BOXCAR2.SML provee un ejemplo más complejo de una ventana de diálogo incorporando una vista de despliegue.



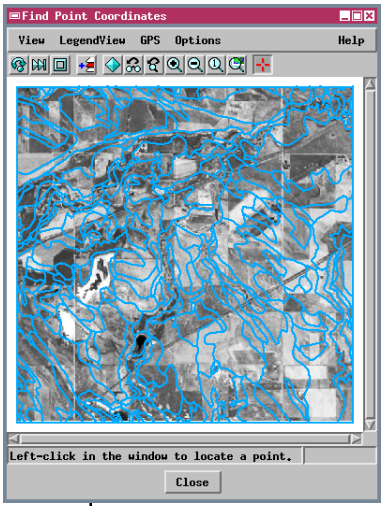
Por defecto, un widget de una vista de despliegue, incluye menúes estándar, barras de herramientas básicas, línea de escala/posición, y línea de estado. El parámetro createflag$ de la función GroupCreateView() le permite si así lo desea eliminar los elementos seleccionados de la ventana. Por ejemplo, la ventana de despliegue Boxcar no tiene la línea de Escala / Posición o línea de estado.

Sistemas de Coordenadas en las Vistas

Los ejercicios previos han discutido las funciones SML que usan la información de georeferencia de los objetos para convertir información de posición entre las coordenadas del objeto (tales como los números de filas y columnas de un raster) y las coordenadas del mapa. Cuando despliega objetos espaciales en una vista dentro de una ventana de dialogo, varios otros sistemas de coordenadas entran en juego. El guión de ejemplo PTCOORD.SML le ayudará a explorar estos sistemas de coordenadas e ilustra los recursos disponibles para la conversión entre ellos. El guión despliega un raster predeterminado (con coordenadas UTM) y un objeto vector (con coordenadas en Latitud y Longitud) y provee una herramienta gráfica tipo punto, con la cual puede escoger una posición. Cuando aplica la herramienta (clic derecho) la posición del punto es reportada en la ventana de la consola en varios sistemas de coordenadas.

PASOS

- ☑ seleccione File / Open / *.SML File y escoja / LITEDATA / SML / PTCOORD.SML
- ☑ ejecute el guión
- ☑ clic con el botón izquierdo en la ventana para ubicar la herramienta de punto
- ☑ clic derecho para mirar las coordenadas en la ventana de Consola
- ☑ pruebe varias ubicaciones de puntos para mirar como varían los distintos tipos de coordenadas
- ☑ estudie el guión para mirar como se ejecutan las transformaciones de coordenadas
- ☑ Cierre la ventana Find Point Coordinates cuando haya terminado



Una herramienta gráfica usada en la vista retorna las posiciones en *coordenadas de la vista*. Para un grupo simple de una sola vista, las coordenadas de la vista corresponden a las coordenadas de mapa del grupo. El sistema de coordenadas del grupo es determinado inicialmente por la georeferencia del primer nivel agregado al grupo, pero puede ser modificado por el guión restableciendo la clase Proyección para el grupo. Las *coordenadas de pantalla* son las coordenadas del área de dibujo de la vista (en pixeles), donde los objetos son de hecho desplegados. Si desea que el guión dibuje características adicionales en el área de dibujo, las funciones de dibujo requieren de las coordenadas de pantalla. Cada nivel en la vista también tiene las *coordenadas del nivel*, las cuales son las coordenadas del objeto para el objeto en el nivel, tanto como las *coordenadas del mapa del nivel.*

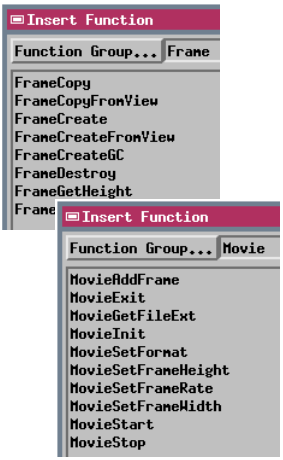
El grupo de funciones Geodata Display View incluye funciones para trasladar entre coordenadas de vista y pantalla, nivel y de mapa de nivel.

```
View coordinates: x = 633864.05, y = 4730504.92
Screen coordinates: x = 67, y = 266
Raster layer (object) coordinates: x = 82.68, y = 366.80
Vector layer (object) coordinates: x = 326.33, y = 2408.09
Raster layer map coordinates: x = 633886.07, y = 4730470.08
        in Universal Transverse Mercator Zone 13 (W 108 to W 102)
Vector layer map coordinates: x = -103.36, y = 42.72
        in Latitude / Longitude
Group coordinates = View coordinates for group view.
Group coordinate system = Universal Transverse Mercator
```

Guiones de Generación de Animaciones

PASOS

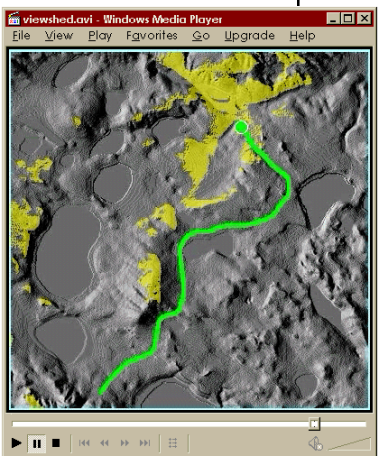
- ☑ seleccione Process / SML / Edit Script del menú principal de TNTmips
- ☑ escoja File / Open / *.SML File y seleccione de su directorio principal TNT CUSTOM / MOVIE / VSHEDMOV.SML
- ☑ estudie la estructura y comentarios del guión



Un guión SML puede crear y registrar animaciones personalizadas desde sus datos geoespaciales. El guión de ejemplo en este ejercicio crea un archivo de animación mostrando una serie de divisorias visuales calculadas desde un raster de elevación en diferentes puntos a lo largo de una línea vector.

Cualquier animación consiste de una secuencia de cuadros estáticos que varían gradualmente. Un guión de generación de animaciones captura los cuadros desde el contenido de una o más ventanas de vista creadas por el guión y copia cada cuadro en un archivo de salida MPEG o AVI. La animación puede por consiguiente registrar cualquier cambio secuencial en la ventana(s) de la vista usada para crear los cuadros. Las funciones en el grupo Frame and Movies son utilizadas para fijar los parámetros generales de cuadro y animación, para capturar el contenido de la ventana de vista a un cuadro y copiar el contenido del cuadro al archivo de salida. Usted puede también anotar cada cuadro con texto o marcas de posición usando las funciones del grupo de funciones de Dibujo.

Los cambios secuenciales en la ventana de vista pueden ser logrados de varias maneras. El guión puede añadir y remover una serie de niveles previamente preparados desde y hacia la vista. También puede modificar los parámetros de despliegue para un nivel continuo único. Para objetos vector, esto podría involucrar acomodar los estilos de elementos en una secuencia de variación de valores de atributos (tales como población en diferentes años). El método final es ejemplificado por el guión VSHEDMOV: el guión mismo calcula los cambios de los datos proporcionados y parámetros. Para cada cuadro en esta animación, el guión calcula la divisoria visual y la despliega en la ventana de vista en amarillo sobre el despliegue sombreado de relieve del modelo de elevación.



Más acerca de guiones para la generación de animaciones esta disponible en un documento en línea en **http://www.microimages.com/relnotes/v64/viewmarks.pdf**

Guiones de Simulaciones 3D

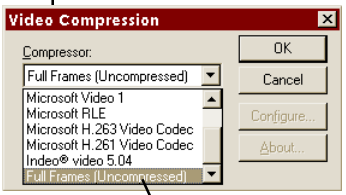
Un guión de animación SML puede utilizar las capacidades de despliegue perspectivo 3D de TNTmips, para registrar animaciones personalizadas 3D. Un guión puede abrir una ventana de vista perspectiva 3D y cambiar los parámetros de despliegue para cada cuadro en la animación, permitiéndole moverse por encima, sobre y alrededor de una superficie 3D. SML incorpora toda la funcionalidad del proceso de simulación 3D de TNTmips, pero expande su control sobre los parámetros de despliegue.

Los miembros de la Clase y métodos de la clase VIEWPOINT3D son usados para manipular los parámetros para la vista 3D. Cada vista 3D tiene una posición del visor y una posición a la que está mirando, el punto donde la vista actual está centrada. SML le provee de un completo control sobre las dos posiciones. Usted puede fijar explícitamente las coordenadas del visor y la posición central de la vista de cada cuadro, o mover cada posición una distancia específica o dirección relativa a la posición previa. Cualquier posición puede ser rotada alrededor de otra. Usted también puede fijar cualquier posición y luego especificar un acimut, ángulo de elevación y distancia para definir la otra posición.

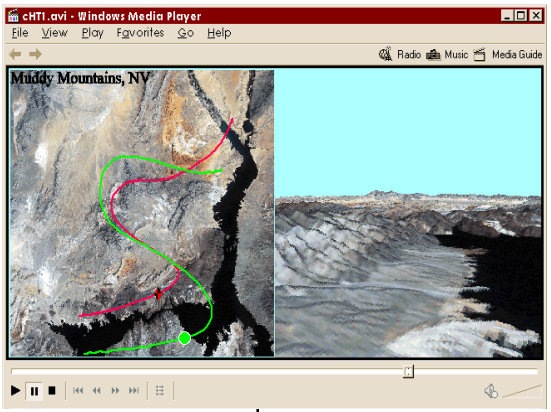
El guión PATHCHT1 copia las dos vistas 2D y 3D en cada cuadro de la animación. Las posiciones del visor y centro de vista son calculadas de líneas vector 2D que están desplegadas en la vista 2D, pero que son ocultadas en la vista 3D. Las posiciones actuales del visor y centro de vista se muestran por medio de símbolos dibujados en la porción 2D de cada cuadro después de que las vistan han sido capturadas.

PASOS

- ☑ escoja File / Open / *.SML File y seleccione de su directorio principal de TNT CUSTOM / MOVIE / PATHCHT1.SML
- ☑ estudie la estructura y comentarios del guión



Para registrar una animación desde un guión SML, se debe disponer de un software capaz de codificar archivos MPEG (cualquier plataforma de computadora) o archivos AVI (únicamente plataformas Windows). Cuando el registro inicia, se abre una ventana que le permite seleccionar las opciones de compresión.



Animaciones creadas desde este ejemplo de guión de animación SML puede obtenerse en

http://www.microimages.com/promo/smlmovies

APPLIDATs

Usted puede usar SML para crear productos llave en mano de aplicaciones geoespaciales autónomos, denominados APPLIDATs. Un APPLIDAT puede incluir un guión SML o una serie de guiones junto con los datos geoespaciales a ser procesados. Dado que se acoplan datos y guiones, ellos se cargan juntos automáticamente cuando el APPLIDAT se ejecuta. No hay necesidad de que el usuario navegue y cargue los datos manualmente. Un APPLIDAT es por consiguiente ideal para proporcionar datos con aplicaciones personalizadas a usuarios que no están familiarizado con la interface de TNT.

Un APPLIDAT incluye uno o más objetos guión SML en un Archivo de Proyecto TNT, que se ha renombrado con la extensión de archivo. SML. Los usuarios pueden ejecutar un APPLIDAT con un doble clic en el archivo o usando un icono de acceso directo del escritorio. (Como se indica en este ejercicio, los usuarios de TNTmips también pueden ejecutar un APPLIDAT del Menú Custom / APPLIDAT.) Ejecutando un APPLIDAT inicializa TNTview (con la interface normal escondida) y abre una barra de herramientas personalizada, con un icono para cada guión incluido. Los botones de iconos para abrir la interface estándar de TNTview y para salir del APPLIDAT también son incluidos automáticamente. Usted puede escribir los componentes del guión para usar datos almacenados en el mismo Archivo de Proyecto SML o en un Archivo de Proyecto normal acompañando en el mismo directorio.

Cuando un objeto guión se crea en un Archivo de Proyecto, TNT le asigna automáticamente un icono de subobjeto predefinido, que usted puede revisar o puede cambiar con un icono diferente. Cuando el APPLIDAT es ejecutado, se agregan a la barra de herramientas los botones de los iconos de los guiones, desde la izquierda en orden alfabético según los nombres de guión. Si su APPLIDAT incluye varios guiones que deben ejecutarse en un orden definido, nombre los guiones para que el orden alfabético de sus nombres siga la sucesión definida del proceso. La descripción de un objeto guión se usa automáticamente como la guía informativa para el botón de su icono.

PASOS

- ☑ seleccione Custom / APPLIDAT / BENCHMRK
- ☑ clic el botón del icono Instructions en la barra de herramientas
- ☑ presione [Close] en la ventana Help
- ☑ clic el botón del icono TNT Benchmark
- ☑ pruebe algunos de los procesos benchmark, luego presione [Exit]
- ☑ clic el botón Exit en la barra de herramientas

Barra de herramientas del APPKIDAT Benchmark

Guión SML: TNT Benchmark

Instrucciones TNTview Salir

- ☑ seleccione File / Open / RVC Object en la ventana SML
- ☑ seleccione /LITEDATA / SML / SMLLAYER.RVC / ARROW
- ☑ seleccione el icono File / Edit Toolbar
- ☑ en la ventana Select Bitmap Pattern, clic el botón Set y escoja el conjunto Advisor de la lista
- ☑ seleccione el icono "gold" ilustrado y clic OK

- ☑ clic [Yes] para confirmar su opción en la caja de dialogo Verify

Proporcionando Instrucciones para APPLIDATs

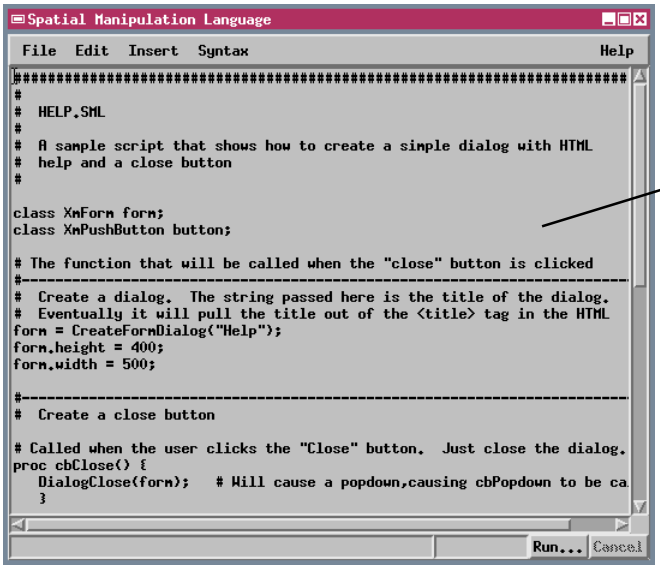
SML le permite escribir APPLIDATs que tienen una interface *intuitiva*. Sus usuarios no necesitan ser entrenados en (o incluso estar conscientes de) los productos TNT. Todas las instrucciones necesarias pueden descubrirse la primera vez que el APPLIDAT se usa, o ser fácilmente redescubiertas después de un lapso de tiempo. Simplemente incluya en su APPLIDAT una copia del guión HELP.SML del APPLIDAT BENCHMARK. Este guión crea una ventana del diálogo para desplegar texto en formato HTML e ilustraciones. El juego de instrucciones HTML se almacena como un subobjeto del guión de HELP.SML.

Un conjunto de instrucciones es fácil de crear y mantener porque usted puede usar cualquier editor que soporte el formato HTML. Así usted puede escribir sus instrucciones en un programa como Microsoft Word y usar la opción Guardar Como... para almacenar el archivo en formato HTML. Para asociar su nuevo archivo de ayuda con el APPLIDAT, edite el guión HELP en el editor de guiones SML y seleccione Add Text Objects del menú File. Cuando usted selecciona su archivo HTML, TNT lo copia a un subobjeto del guión.

NOTA: para abrir los objetos guión en un Archivo de Proyecto APPLIDAT (archivo de extensión .SML) en el editor SML, usted debe usar File / Open / *.SML File. Cuando usted selecciona un archivo SML que en realidad es un Archivo de Proyecto, se abre una ventana de selección de objetos, para permitirle seleccionar un objeto guión desde dentro del archivo.

PASOS

- ☑ escoja File / Open / *.SML File en la ventana SML
- ☑ seleccione BENCHMRK.SML en la ventana Select File
- ☑ seleccione el objeto guión Help en la ventana Select Object
- ☑ examine la estructura y comentarios del guión



Usted puede copiar este guión de Ayuda a su propio archivo APPLIDAT y usarlo directamente para crear sus Instrucciones o ventana de Ayuda. Un juego de instrucciones no se separará de su APPLIDAT porque está empaquetado con los otros recursos.

APPLIDAT BIOMASS2

PASOS

- ☑ seleccione Support / Maintenance / Project File del menú principal de TNTmips y examine el contenido de BIOMASS2.SML en la carpeta Custom / APPLIDAT de su carpeta principal de productos TNT
- ☑ salir de Project File Maintenance y seleccione Custom / APPLIDAT / BIOMASS2
- ☑ clic el botón del icono Instructions y lea las instrucciones
- ☑ clic en el botón del icono Biomass Mapping, defina un área para mapear, filtre el resultado, y conviértalo a vector
- ☑ salga de la ventana Biomass Mapping
- ☑ ejecutar las aplicaciones de Asset Mapping y 3D Simulation
- ☑ salga del APPLIDAT BIOMASS2 cuando haya terminado

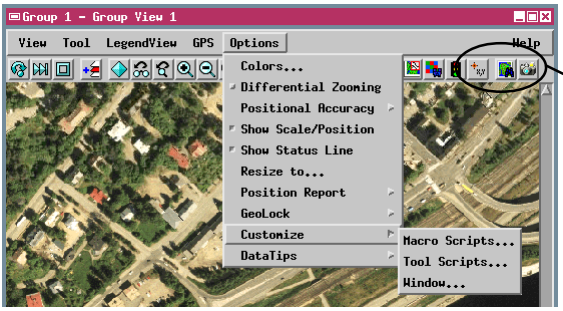
El APPLIDAT biomass2 fue escrito por MicroImages para proporcionar un ejemplo y prototipo de un producto llave en mano de APPLIDAT. Este ilustra cómo un APPLIDAT puede permitir al usuario llevar a cabo una serie de operaciones en los datos de la entrada y automáticamente pasar a los productos intermedios a lo largo de la siguiente operación. En este ejemplo la aplicación le permitiría a un granjero determinar la biomasa de la cosecha para cualquiera área designada de una imagen infrarroja en color, desplegar los recursos de la granja encima de la imagen y mapa de la biomasa, y despliega una vista perspectiva 3D de la imagen y mapa de la biomasa. Las Instrucciones para el APPLIDAT BIOMASS2 proporcionan una apreciación global más detallada de cada operación.

El archivo APPLIDAT (BIOMASS2.SML) incluye tres objetos guión de procesamiento: Biomass (Mapeo de Biomasa), Pinmap (Despliegue de Recursos), y View3D (Simulación 3D) esto está diseñado para ser ejecutado en ese orden (note el orden alfabético de los nombres de los guiones y las posiciones de sus iconos en la barra de herramientas). Instrucciones para el producto están contenidas en el guión llamado About (note que el propio guión contiene las instrucciones en formato HTML, en lugar de usar un subobjeto HTML). Todos los datos de entrada están en el archivo APPLIDAT. Los objetos espaciales producidos por el APPLIDAT, se almacenan y recuperan conforme se necesitan en el Archivo de Proyecto adjunto BIOMASS.RVC.

Después de que usted ha ejecutado el APPLIDAT, debe examinar la estructura y componentes del guión. Cada guión contiene el código para crear su ventana de diálogo y controles, los procedimientos de llamadas asignados a esos controles, e instrucciones para la entrada y salida de datos. Usted puede usar éstos como modelos para desarrollar sus propios programas de APPLIDAT tipo llave en mano.

Guiones de Herramientas y Macros

Guiones de Herramientas y Macros son formas especializadas de guiones SML que se inician desde un botón del icono en la ventana de Despliegue y se acceden automáticamente y operan en los objetos en la vista. Usted puede crear guiones de herramientas o guiones de macros que permiten a cualquier usuario ejecutar los procedimientos personalizados en las capas de datos espaciales cargadas en la ventana de despliegue. Después de que usted agrega un guión de herramientas o un guión de macros, el botón del icono aparece en la barra de herramientas de cada ventana de despliegue para todos los procesos de TNT. Y cada ventana View ofrece selecciones de menú que le permiten fácilmente agregar o eliminar guiones de Herramienta y guiones de Macros (Options / Customize).



Los guiones de macros y herramientas son ejecutados desde los botones de los iconos en la barra de herramientas de la ventana de despliegue.

Para el desarrollador de guiones, los guiones de macros y de herramientas le proporcionan una manera aerodinámica de proporcionar capacidades de procesamiento personalizadas, que requieren la interacción visual con los datos espaciales. Para hacer esto en un guión SML normal, usted tiene que proporcionar el código para crear y manejar la ventana View y su contenido. Pero porque se invocan guiones de macros y de herramientas desde una ventana de despliegue, la mayoría de esa gestión es ejecutada automáticamente, y usted puede enfocarse en codificar el procesamiento personalizado propiamente dicho.

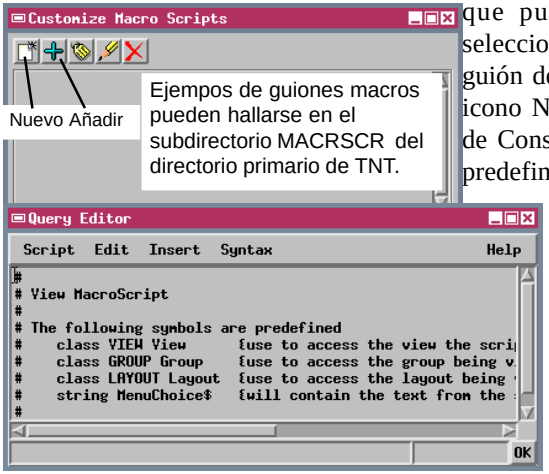
Los guiones de macros y de herramientas:

- se ejecutan desde un botón del icono en la barra de herramientas de la ventana de Vista;
- pueden acceder elementos de la vista actual, como los niveles, extensión, proyección, elementos seleccionados, factor de ampliación, escala, y estilos;
- pueden operar en los objetos en la vista actual u objetos contenidos en la misma área;
- pueden agregar un nivel recientemente creado a la vista;
- pueden empezar un programa externo y proporcionarle datos derivados de la vista actual.

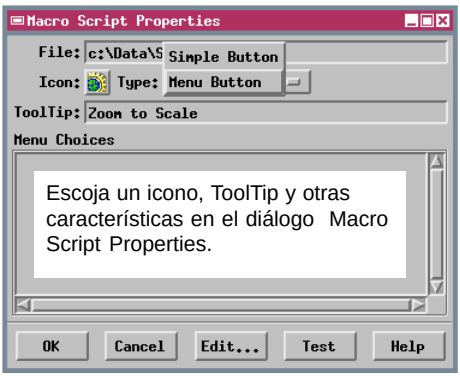
Un guión de herramienta invoca una herramienta de dibujo y/o una ventana de diálogo (definida por el desarrollador del guión), eso le permite al usuario interactuar con los datos espaciales en la ventana de vista. Por ejemplo, el usuario podría delinear una área o seleccionar los elementos particulares ser procesados. Un guión de macros no permite tal interacción gráfica, pero puede establecerse con un menú desplegable que proporciona las opciones del programa.

Estructura de un Guión de Macros

Para agregar un guión de macros de forma que pueda ejecutarse desde un icono en la barra de herramientas de la ventana View, escoja Options / Customize / Macro Scripts de la ventana View en cualquier proceso. Haciendo esta selección se abre la ventana Customize Macro Scripts. Si quiere agregar un guión existente, pulse el botón del icono Add para abrir la ventana de selección de archivo, para que pueda navegar al guión y seleccionarlo. Para crear un nuevo guión de macro, pulse el botón del icono New. Una ventana de Editor de Consultas se abre con un guión predefinido que contiene una lista de símbolos predefinidos que puede usar en el guión de macros. La ventana de Editor de Consultas incluye todas las características de creación y edición de guiones de una ventana SML normal.



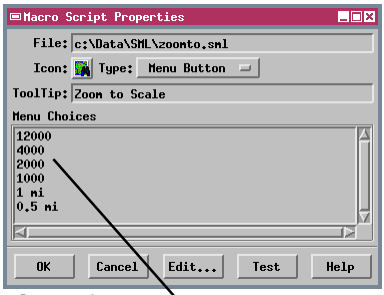
Una vez usted ha creado o ha agregado un guión de macros, la ventana Macro Script Properties se abre. Esta ventana le permite escoger un icono, indicar si el guión se ejecuta de un botón simple o un botón del menú, prepara la barra de herramientas para el botón del icono, ingresa los ítems del menú si se utiliza un botón de menú, y prueba su guión. Escoja un botón simple para hacer que su guión de herramienta se ejecute automáticamente sin ingreso adicional del usuario. Escoja un botón de menú si quiere que se presenten opciones desplegables cuando el botón es pulsado. Si es escogido un Botón de Menú en la ventana Macro Script Properties, el campo de texto Menu Choices se pone activo para que usted pueda ingresar en las opciones del menú necesarias para el guión.



Ingrese la información que quiere directamente en el campo ToolTip. Esta información aparece cuando el cursor se coloca encima del icono del guión de macros en la ventana de Vista. El botón de Prueba al fondo de la ventana le permite ejecutar su guión sin cerrar las ventanas personalizadas. Pulse OK en las ventanas Macro Script Properties y Customize Macro Scripts cuando haya terminado de añadir, desarrollar, y/o comprobar su guión.

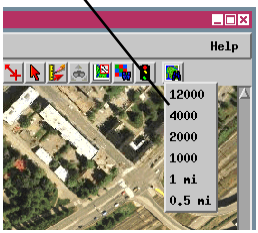
Ejemplo de Guión de Macros: Ampliación a Escala

Algunos ejemplos de guiones de macros se proporcionan en el subdirectorio MACRSCR en su directorio primario de TNT. Estudie estos ejemplos para entender como estructurar sus propios guiones de macros.

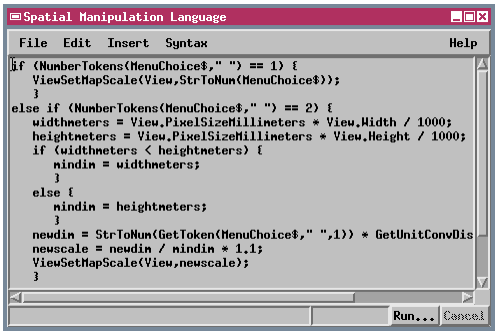


El guión de macros Zoom to Scale le permite al visor redibujar la ventana de Vista a una de varias escalas de mapas seleccionadas del botón del menú desplegable del guión. Para una operación adecuada del guión, los objetos en la ventana de vista o deben estar georeferenciados, o deben estar calibrados a una escala.

Al instalar el guión Zoom to Scale, fije las opciones de menú de escala que sean las más apropiadas para sus datos espaciales.



Las selecciones del menú no se predeterminan por el guión Zoom to Scale. Cuando usted instala el guión, es libre de preparar las opciones del menú con el rango de selecciones de escala más apropiado para sus datos. El guión acepta ingreso de escala del menú como escala del mapa o dimensiones del terreno. Si la entrada del menú es completamente numérica, se interpreta como el denominador de la fracción de la escala de mapa. Por ejemplo, 12000 es interpretado como una escala del mapa de 1:12000. Si el ingreso del menú está en dos partes separadas por un espacio (como “1 mi”), la primera parte de la entrada se interpreta como una dimensión en millas del terreno. (Esta porción del guión puede modificarse fácilmente para aceptar las dimensiones en kilómetros u otras unidades de distancia.) El guión realiza luego los cálculos necesarios y fija entonces la nueva escala del mapa para la ventana de Vista.

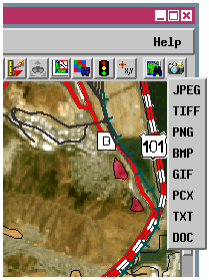


La variable predefinida del guión de macros MenuChoice$ es usada para representar la selección del usuario en el botón del menú del guión de macros. Para la entrada numérica, este texto debe convertirse a un valor numérico usando la función StrToNum ().

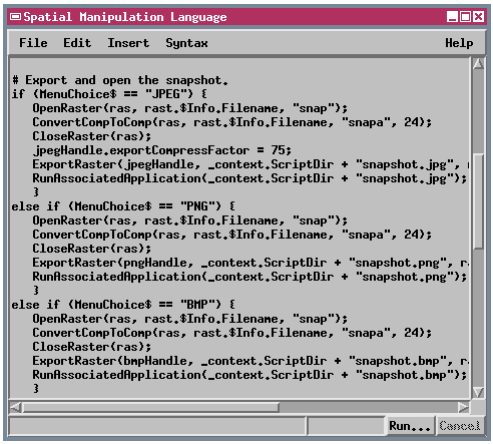
Más acerca del guión de macro Zoom to Scale está disponible en el documento en línea en

http://www.microimages.com/relnotes/v64/zoomto.pdf

Ejemplo de Guión de Macros: Instantánea

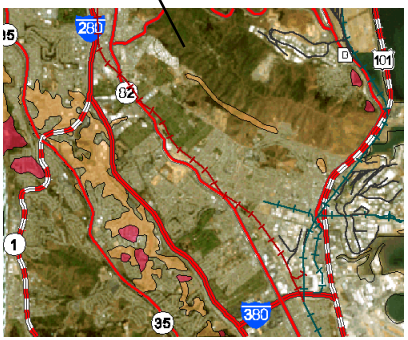


El guión de la Instantánea es un ejemplo simple de un guión de macros que procesa los datos de una ventana de Vista y ejecuta una aplicación externa. El guión captura una instantánea de la pantalla de la ventana de vista y la exporta a un formato de archivo de imagen que usted ha escogido del botón del menú desplegable del guión. El guión lanza el programa de aplicación que usted ha registrado previamente con su sistema operativo para abrir entonces ese tipo de archivo.



El guión de la Instantánea se ha escrito para crear formatos de archivo específicos: JPEG, PNG, BMP, PCX, GIF, TIFF, y archivos ASCII con TXT o DOC como extensiones. Cuando usted agrega este guión de macros a una ventana de Vista, usted debe fijar las opciones para el botón del menú de guión de este conjunto de formatos. El texto para cada entrada del menú debe emparejar exactamente el carácter de texto esperado por el guión, incluyendo mayúsculas y minúsculas (por ejemplo, JPEG en lugar de Jpeg).

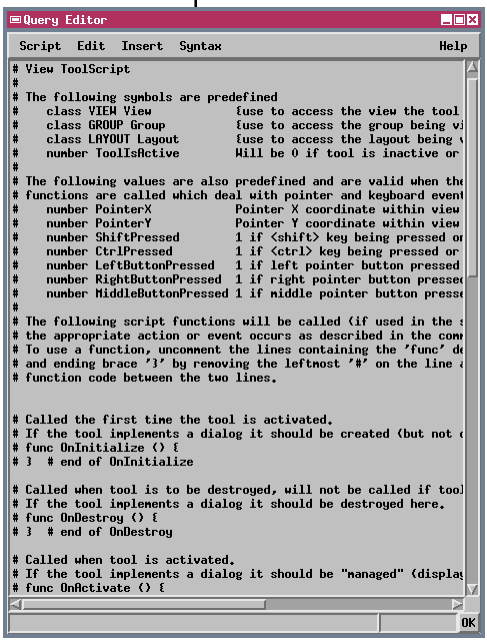
La instantánea almacenada de la ventana de Vista con el fondo del raster y varias sobrepuestas de vector.



El guión almacena la instantánea inicialmente como un objeto raster de color compuesto temporal. La profundidad de bits del color compuesto es determinada por los parámetros de despliegue de su computadora. El segmento de guión para cada formato de archivo realiza una conversión de color a la profundidad de color apropiada para ese formato previo a su exportación. El archivo de salida se almacena automáticamente en el mismo directorio del guión, entonces la aplicación asociada al archivo es ejecutada. Estas operaciones hacen uso de la variable clase _context, la que especifica la información del contexto interior para el guión. El miembro de la clase _context.ScriptDir especifica el directorio en que el guión se encuentra.

Plantilla para Guión de Herramientas

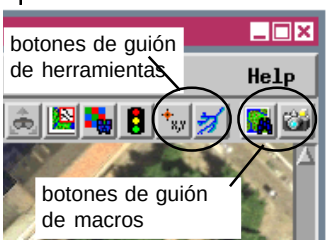
Para añadir un guión de herramienta que ejecute desde el botón de un icono en la barra de herramientas de la ventana de Vista, escoja Options / Customize / Tool Scripts de la ventana View en cualquier proceso de TNT que tiene una ventana de despliegue. Haciendo esta selección se abre la ventana Customize Tool Scripts, que es casi idéntica a la ventana Customize Macro Scripts, tratada previamente.



Para crear un nuevo guión de herramienta, pulse el botón en el botón del icono New para abrir la ventana de Edición de Consultas, la que muestra la plantilla del guión de herramientas. La plantilla lista varios símbolos predefinidos y valores que usted puede usar en cualquier guión de herramientas. Los valores predefinidos incluyen las coordenadas X , Y del cursor de la pantalla dentro de la vista (en pixeles) y valores que graban las acciones de botón del mouse.

Adicionalmente, la plantilla del guión de herramientas incluye las definiciones básicas de las funciones probables a ser usadas en un guión de herramientas. Éstos incluyen las funciones usadas la primera vez cuando una herramienta es activada; cuando la herramienta se cancela; cuando la herramienta es activada y desactivada; cuando la herramienta se suspende (durante un redibujo) y reasumida (después del redibujo); cuando los botones izquierdo, derecho, o medio del mouse se presionan o sueltan; cuando el cursor se mueve sin presionar un botón; cuando el cursor se mueve con un botón presionado; cuando el cursor entra o deja la ventana de despliegue; y cuando el usuario presiona una tecla. Para crear su guión, remueva los caracteres de comentario (#) a la izquierda de cada definición de la función usted necesita y agregue el código para especificar la acción deseada a ser llevada a cabo por esa función.

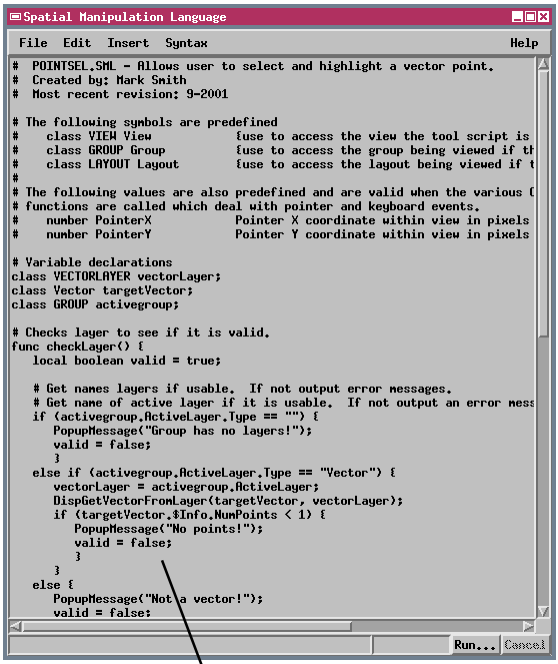
Los botones del icono del guión de herramientas, aparecen a la izquierda de cualquier icono de Guión de Macros en la barra de herramientas de la ventana de Vista.



Ejemplo de Guión de Herramientas: Selección de Puntos

Se proporcionan varios ejemplos de guiones de herramientas con los productos TNT en el subdirectorio TOOLSCR bajo su directorio principal de TNT. Usted puede usar los componentes de cualquiera o todos estos guiones para crear la herramienta personalizada que usted necesita para su aplicación especializada.

El guión de selección de punto (POINTSEL.SML) ilustra cómo preparar una guión de herramientas que permite al usuario interactivamente seleccionar elementos de un objeto vector en la ventana de despliegue. En este caso el guión selecciona el elemento punto más cercano cuando el botón del ratón izquierdo se presiona; esta acción se controla por la definición de la función OnLeftButtonPress (). Este simple guión únicamente selecciona el punto, pero la función de presionar el botón podría expandirse para usar el punto seleccionado para procesos posteriores, como escribir a un archivo externo las coordenadas de mapa de cada punto.

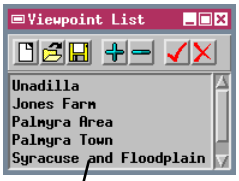


Los Guiones de Herramientas pueden incluir procedimientos definidos por el usuario y funciones que se llaman por otras funciones en el guión.

Dado que un guión de herramientas se ejecuta interactivamente desde una ventana de despliegue, todo el proceso se lleva a cabo por funciones del guión ejecutadas por las acciones del mouse o por acciones llevadas a cabo en ventanas de diálogo creadas por el guión. Las definiciones de la función que usted proporciona para los nombres de las funciones predefinidas, pueden llamar a otras funciones y procedimientos definidos en otras partes del guión de herramientas. En el guión de selección de puntos, por ejemplo, la función OnLeftButtonPress() llama a una función previamente definida checkLayer()que verifica para asegurarse que el grupo activo contiene un nivel, y que el nivel es un objeto vector. La función OnInitialize también llama a un procedimiento cbGroup() para identificar al grupo activo en un formato de múltiples grupos. Este código generaliza el guión de herramientas para el uso bien sea en una vista de grupo o en una ventana de vista de formato.

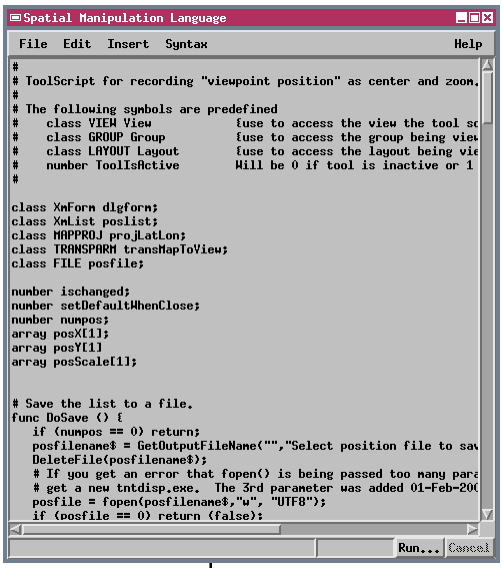
Ejemplo de Guión de Herramientas: Marcas de Vista

El guión de herramientas de (Marcas de vista) ViewMarks (VPTOOL.SML) le permite registrar una lista de marcas de posición para la ventana de Vista. Un ViewMark registra las coordenadas de mapa del centro de vista actual (en latitud/longitud) y la escala del mapa. Una vez creada la lista, usted puede seleccionar un ViewMark y centrar la ventana de despliegue en esa posición y escala designadas. Los ViewMarks son particularmente útiles para formatos que cubren una área geográfica grande, sobre todo cuando el formato usa la limitación de visibilidad por escala del mapa para agregar y quitar niveles automáticamente, conforme usted amplia o reduce el área de vista en la pantalla de despliegue.

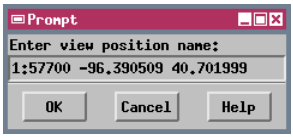


VPTOOL.SML le permite escoger un punto de vista de la Lista de puntos, para centrar la vista a esa ubicación y escala.

El guión de ViewMarks crea una ventana de diálogo Viewpoint List que proporciona una lista interactiva así como los botones usados para inicializar las acciones del guión; no hay ninguna herramienta gráfica creada por el guión. Este diálogo se crea por medio de la función OnInitialize (). Los botones del icono en la ventana le permiten agregar o quite ViewMarks de la lista y hacer un acercamiento a la marca seleccionada. Otros botones para presionar le permiten guardar la lista a un archivo de texto, abrir un archivo existente de puntos de vista, crear una nueva lista, o cerrar la ventana. Cada uno de estos botones llama a una función separada o a un procedimiento definido en el guión de herramientas.



Cuando usted agrega un ViewMark, una ventana de ingreso se abre para permitirle nombrar la marca. (El nombre predefinido es el nivel de ampliación y coordenada de la posición). Los nombres de ViewMark se guardan en una lista de widget (clase XmList). Los valores de la escala y las coordenadas X, Y se almacenan en arreglos numéricos separados.

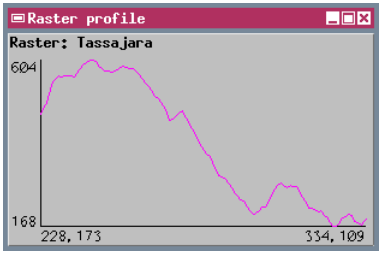


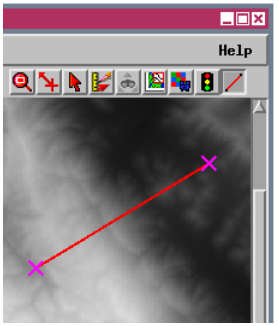
Más acerca del guión de herramienta ViewMarks esta disponible en un docuemnto en línea en

http://www.microimages.com/relnotes/v64/viewmarks.pdf

Ejemplo de Guión de Herramientas: Perfil Raster

El guión de herramientas para Perfil Raster (RASTPROF.SML) proporciona una herramienta de línea que registra y plotea un perfil de los valores raster de las celdas a lo largo de una línea dibujada por el usuario. El raster designado para el perfil deben ser la capa activa en la vista, y las posiciones x-y para los valores se registran en las coordenadas del raster (la columna y fila de la línea). Aunque el ploteo del perfil es el resultado final en este ejemplo, el guión puede modificarse para convertir las posiciones a coordenadas del mapa, aplicar un procesamiento adicional a los valores del perfil, o escriba éstos a un archivo de texto.

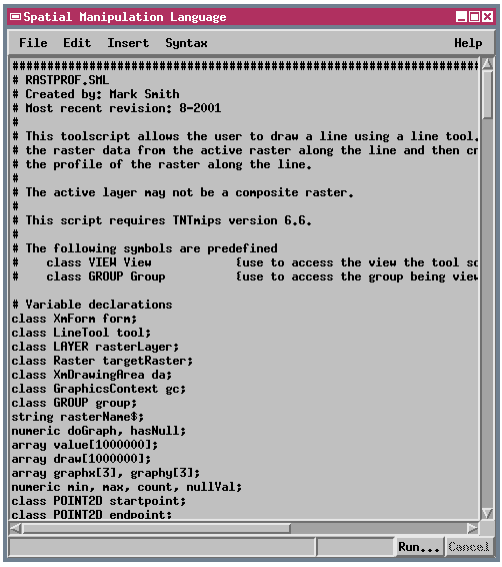




Una porción de la función OnInitialize () en el guión invoca una herramienta estándar interactiva de línea:

```
tool = ViewCreateLineTool(View);
ToolAddCallback(tool.ApplyCallback,
                cbToolApply);
```

(La herramienta variable fue previamente declarada como un miembro de la clase LineTool.) El procedimiento cbToolApply() que adquiere el perfil se llama cuando la herramienta es aplicada por medio de una presión del botón derecho del ratón.

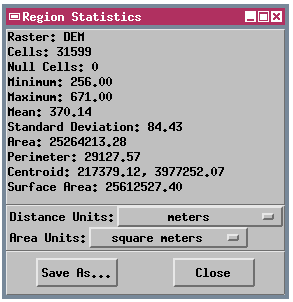


Este enlace se establece en la segunda declaración del fragmento arriba, el mismo que añade el nombre del procedimiento a la lista de la herramienta ApplyCallback. Esta estructura exime de la necesidad de una función OnRightButtonPush separada.

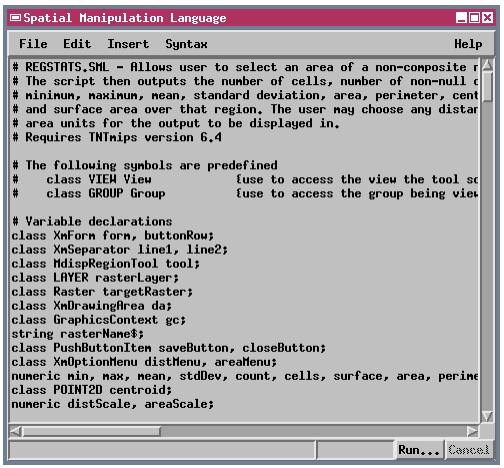
El guión también demuestra cómo el resultado de una acción puede mostrarse gráficamente en una ventana creada por el guión. El código que dibuja los ejes del gráfico, las etiquetas, y el perfil esta contenido en el procedimiento cbRedraw() definido en el guión.

Guión de Herramientas: Estadísticas de un Área

El guión de herramienta de Estadísticas de Area (REGSTATS.SML) muestra cómo usted puede crear una herramienta personalizada para permitir al usuario dibujar un polígono en la ventana de despliegue, convertir el polígono a una región, y usar la región para operar en otro objeto. En este ejemplo, la región se usa para la tarea simple de extraer las estadísticas de un nivel raster en la vista. Pero el guión podría modificarse para ejecutar muchas otras funciones, como crear una máscara raster o para extraer los elementos de un objeto vector. Las operaciones de la región no se restringen a las capas en la vista; usted puede operar en cualquier objeto georeferenciado que traslape la región definida.



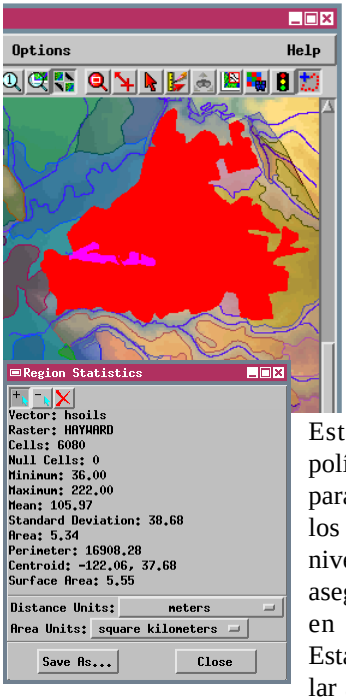
Este guión opera en un objeto raster que es la capa activa en la vista. En el ejemplo mostrado aquí, el polígono es dibujado sobre un nivel de una imagen traslapando el nivel activo que contiene un raster de elevación. Usando la región definida por la herramienta del polígono, el guión calcula el número de celdas, de celdas nulas, mínimo, máximo, media y desviación estándar de los valores raster incluidos, y el área, perímetro, ubicación del centroide, y área de la superficie de la región. (Las estadísticas pueden calcularse para cualquier tipo de raster en escala de grises o binario, pero no para rasters compuestos o niveles de raster RGB) Las estadísticas se muestran en una ventana de diálogo Estadísticas de la Región creada por el guión. El guión puede convertir los valores de distancia y área a las unidades seleccionadas del menú de opciones en la ventana. Las estadísticas también pueden almacenarse a un archivo de texto.



Más acerca del guión Estadísticas de un Área está disponible en el documento en línea en

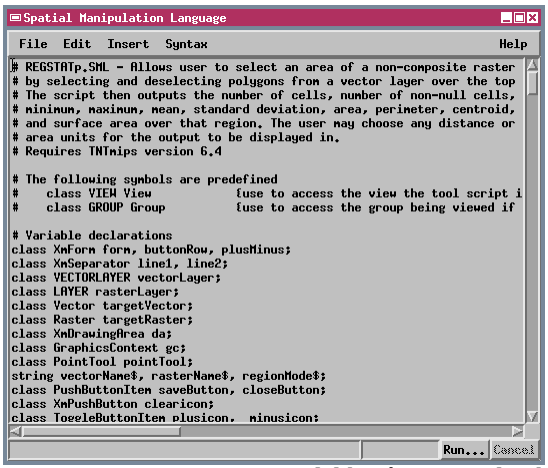
http://www.microimages.com/relnotes/v64/polystats.pdf

Ejemplo de Guión: Estadísticas de una Región



El guión de herramientas de Estadísticas de una Región (REGSTATP.SML), demuestra el diseño para una guión que permite al usuario seleccionar los polígonos de la ventana de vista, crea una región a partir de los polígonos seleccionados, y usa la región resultante para realizar una acción en otro objeto. La tarea del ejemplo para este guión es igual que para el guión de las Estadísticas de un Área: calcula las estadísticas de un nivel del raster en la vista. Como ese guión, sin embargo, usted podría volver a escribir el procedimiento cbToolApply () para realizar diferentes tipos de operaciones en otros objetos.

Este guión le permite seleccionar uno o más polígonos de la capa superior en la vista (y verifica para asegurar que ese nivel es un objeto vector con los polígonos). Se calculan las estadísticas para el nivel del fondo en la vista; el guión verifica para asegurarse que ese nivel contiene un objeto raster en escala de grises o binario. La ventana de Estadísticas de Región creada por el guión es similar al usado por el guión de Estadísticas de Área, pero incluye en el borde superior botones para presionar, que permiten al usuario indicar si el polígono seleccionado debe agregarse o substraerse de la región, y un botón para despejar la región.



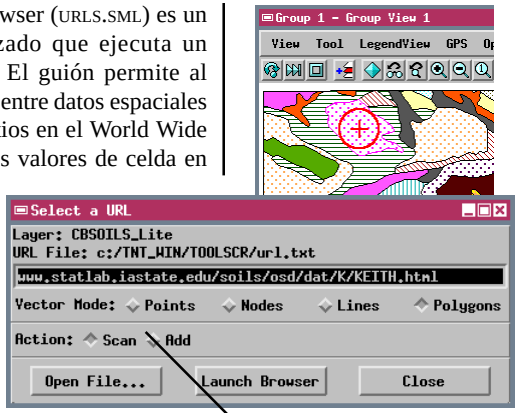
El guión de Estadísticas de Región invoca una herramienta estándar de punto con acciones del botón del mouse predefinidas. Una presión del botón izquierdo ubica la herramienta de punto, y una presión del botón derecho selecciona los polígonos encerrados por la región.

Más acerca del guión de Estadísticas de Región esta disponible en el documento en línea en

http://www.microimages.com/relnotes/v64/regionstatistics.pdf

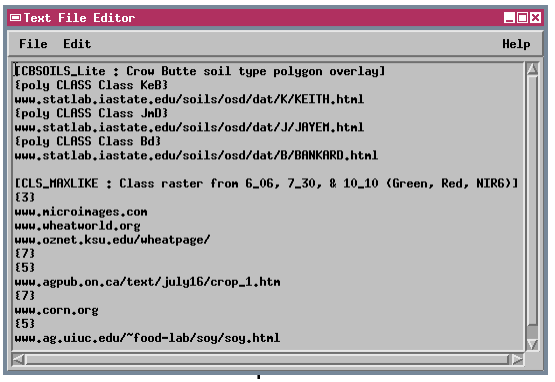
Ejemplo de Guión: Ejecutar un Explorador

El guión de herramientas Run Browser (URLS.SML) es un ejemplo de un guión personalizado que ejecuta un programa de aplicación externo. El guión permite al usuario preparar y usar los enlaces entre datos espaciales en una ventana de despliegue y sitios en el World Wide Web. Pueden hacerse enlaces a los valores de celda en un raster, o a valores de atributo específicos asociados con los elementos del vector. En uno o más URLs pueden asignarse para cada valor. Una vez que los enlaces han sido establecidos, el usuario puede seleccionar un elemento o celda en la ventana de despliegue, escoger el URL deseado, y entonces el guión ejecuta el explorador de red (Web Browser) predefinido, el cual va a la dirección Web deseada.



Active el botón Add para fijar los enlaces, y el botón Scan para usar los enlaces existentes. Para usar los enlaces en modo Scan, seleccione su URL deseado y haga clic [Launch Browser].

Para usar la herramienta, hacer un clic con el botón izquierdo en el polígono o la celda deseada, luego un clic en el botón derecho para confirmar que la herramienta de selección está posicionada correctamente. El URL(s) asociado con el rasgo seleccionado aparece en la ventana Select a URL que es creada por el guión. Escoja el URL deseado, luego haga clic en el botón de Lanzamiento del Navegador.



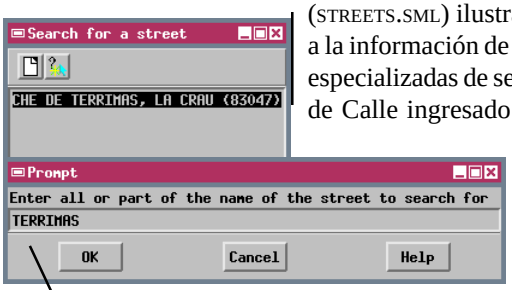
Los enlaces entre URLs y atributos del elemento o valores de la celda, se almacenan en un archivo del texto separado, especificado en el guión de ejemplo nombrado como URL.TXT. El archivo de texto lista el nombre y descripción para cada objeto con los enlaces de URL. Las asociaciones en este ejemplo de guión de herramientas se refieren específicamente a CB_DATA / CB_SOILS.RVC / CBSOILS_LITE, o BEREA / BERCRPCL.RVC / CLS_MAXLIKE.

Más información acerca del guión Run Browser está disponible en el docuemnto en línea en

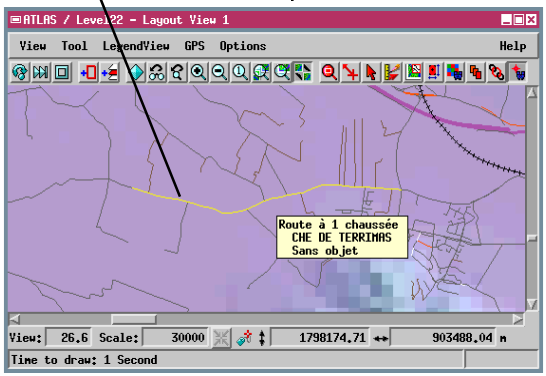
http://www.microimages.com/relnotes/v64/runbrowser.pdf

Ejemplo de Guión: Búsqueda de Calles

El guión de herramientas para Búsqueda de Calles (STREETS.SML) ilustra cómo un guión puede acceder a la información de la base de datos y realizar tareas especializadas de selección. El guión usa un nombre de Calle ingresado por el usuario para localizar y resaltar la línea vector que representa la Calle. El usuario puede ingresar todo o parte de un nombre de Calle, y el guión de herramientas despliega una lista de todas las calles que contienen ese texto de la búsqueda. Cuando el usuario escoge una Calle de la lista, el guión vuelve a dibujar la vista a 1:30000 con todas las líneas que forman parte de la Calle resaltadas y centradas en la Vista. Si las líneas de toda la Calle no encajan en la Vista a 1:30000, la Vista se vuelve a dibujar a una escala que contiene totalmente las líneas.



El usuario ingresa un nombre de Calle y el guión de herramientas lo encuentra en el mapa.



El guión usa los colores actuales para realce de los elementos activos y seleccionados (Options / Colors). Para esta herramienta, la Calle seleccionada tendrá una apariencia uniforme si tanto el color activo como seleccionado son el mismo (amarillo en la ilustración de la ventana).

El nombre de la ciudad y el código postal también se proporciona en la lista de calles encontrada. El guión asume que no hay dos calles separadas con el mismo código postal y con el mismo nombre. Si, sin embargo, resulta que el nombre de la búsqueda pertenece a dos calles diferentes en el mismo código postal (una Calle Principal, y otro Paseo Principal, por ejemplo), se lista sólo el primero encontrado pero los dos se resaltan cuando esta selección se ejecuta.

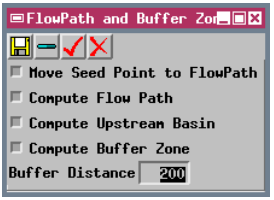
STREETS.SML está codificado para trabajar con geodatos específicos de un atlas de ejemplo de Francia. Usted debe modificar el guión antes de que funcione con otros geodatos y atributos.

Más acerca del guión de herramientas Find Streets está disponible en el docuemnto en línea en

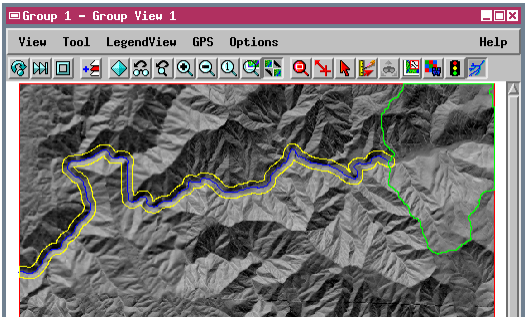
http://www.microimages.com/relnotes/v64/findstreets.pdf

Ejemplo de Guión: Camino de un Flujo

El guión de herramienta del Camino de un Flujo (Flow Path) muestra cómo pueden realizarse procedimientos de análisis personalizados sobre niveles en la vista actual, usando un Guión de Herramienta SML. El guión usa funciones SML watershed que operan en un raster de elevación (DEM) que debe ser la primera capa en la ventana de despliegue.



Cuando el usuario inicia el guión, primero ejecuta funciones de watershed para crear una versión sin depresiones del DEM y un juego completo de vectores de los caminos de flujo. Estas características derivadas se requieren para los pasos subsecuentes del guión; ellos se guardan como objetos temporales y no se despliegan en la vista. El guión abre luego una ventana de FlowPath y Zona de Influencia y crea una herramienta gráfica que permite al usuario poner uno o más puntos semilla de la cuenca (watershed seed points) sobre el DEM o en un nivel de imagen que se traslape. Los botones interruptor en la ventana le permiten al usuario escoger calcular y desplegar:



• la cuenca upstream – aguas arriba- (el área con flujo hacia el punto semilla),
• el camino del flujo downstream -aguas abajo- , y
• una zona de influencia alrededor del camino de flujo.

Si el usuario pretende que los puntos semilla caigan a lo largo de un curso del arroyo, puede activar el botón interruptor Move Seed Point to Flow Path. Cada punto semilla se mueve entonces al camino de flujo precalculado más cercano antes de que el nuevo camino de flujo y cuenca sean calculados. El usuario puede colocar nuevos puntos semilla y repetir el análisis tantas veces como desee, y almacenar los objetos vector calculados.

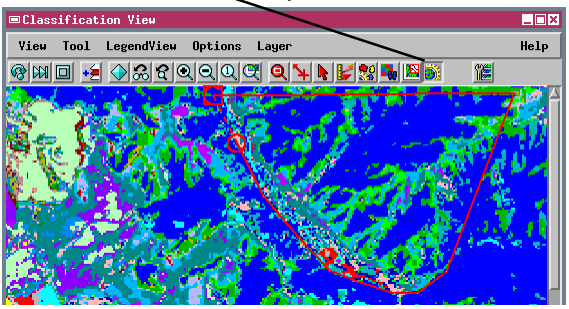
El guión también crea y despliega (en rojo) un nivel del vector que demarca la extensión del DEM. Si un nivel de imagen con traslapo es más grande que el DEM, el usuario puede usar la caja de extensión para guiar la colocación de los puntos semilla. La caja de la extensión también se usa para automáticamente cortar zonas de influencia calculadas a partir de los caminos de flujo que intersectan el límite del DEM.

Más acerca del guión de herramienta Flow Path está disponible en el docuemnto en línea en

http://www.microimages.com/relnotes/v64/flowpath.pdf

Ejemplo de Guión: FRAGSTATS

Una vez instalado un guión de herramientas, puede ejecutarse desde cualquier ventana de despliegue. Así que usted puede ejecutar el proceso de Clasificación Automático e inmediatamente puede ejecutar el guión de herramienta Fragstats en la parte del raster de Clase que se muestra en la ventana de despliegue de la Clasificación.



Una guión separado para ejecutar FRAGSTATS desde la interface de procesos SML también está disponible. FRAGSTAT.SML puede encontrarlo en su directorio principal de TNT bajo / Custom / General. Este guión requiere que usted proporcione los raster de clase y una máscara raster binaria para definir el área de interés.

El guión de herramientas FRAGSTATS (FRAGTOOL.SML) es un ejemplo de un guión que extrae los datos espaciales de un nivel raster en el despliegue y los pasa a una aplicación de un programa externo para procesarlos. El programa de FRAGSTATS fue desarrollado por ecólogos del paisaje para calcular una variedad de estadísticas sobre los modelos espaciales de áreas (parches) representando diferentes clases de hábitats ecológicos. La entrada apropiada para la herramienta es por consiguiente una clase raster que tenga un único valor entero asignado a las celdas para cada categoría o clase. Usted puede crear el rasters de clase a partir de imágenes multiespectrales usando los procesos de Clasificación Automática o el Mapeo de Características en TNTmips.

El guión de herramienta FRAGSTATS proporciona una herramienta del polígono que permite al usuario seleccionar un área (creada como un objeto región temporal) para calcular las estadísticas del paisaje. Cuando la herramienta es aplicada, el guión escribe el raster de clase a un archivo de texto para el uso por el programa FRAGSTATS. A las celdas fuera de la región de interés se les asigna valores de clase negativos en el archivo de texto, que es la convención de FRAGSTATS para identificar celdas que están fuera del “límite del paisaje.” El guión ejecuta entonces el programa FRAGSTATS en un ambiente de DOS. FRAGSTATS identifica los parches homogéneos y calcula las estadísticas para los parches individuales y para las clases completas. Las estadísticas se almacenan en una serie de archivos de texto.

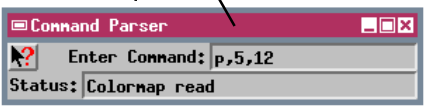
Más acerca del guión de herramientas Fragstats esta disponible en el documento en línes en

http://www.microimages.com/relnotes/v65/fragstats.pdf

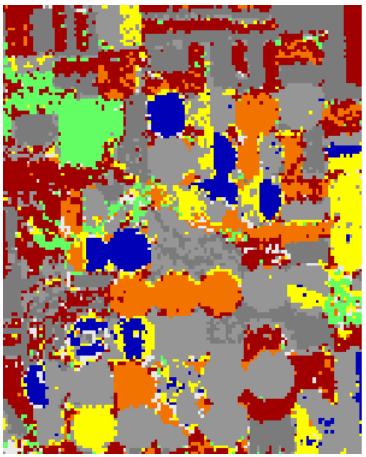
Ejemplo de Guión: Analizador de Comandos

Algunas de los guiones de herramientas discutidos previamente crean una ventana de controles que permite a los usuarios ejecutar acciones del guión usando botones para presionar u otros controles de interface gráficos. El guión de herramientas Analizador de Comandos (COMPAR.SML) demuestra un diseño de guión que crea una interface de "línea de comandos" para ejecutar las acciones del guión. La ventana de Command Parser creada por el guión incluye un campo de texto en el que el usuario ingresa las órdenes de texto predefinidas. Un procedimiento llamado ParseCommand () asocia cada texto de comando con una función particular o el procedimiento definido en otra parte del guión.

La ventana del Evaluador de Comandos creado por el guión incluye un campo para ingresar comandos texto y uno que despliega los mensajes del estado del proceso. Un botón del icono abre una ventana de diálogo de Ayuda.



Este guión de muestra se diseñó como una línea de comandos equivalente al Editor gráfico de la Paleta de Colores en TNTmips. Le permite a un usuario crear o editar una paleta de colores asignando los colores a valores particulares de las celdas o a los rangos de valor de las celdas en un raster. El guión usa un juego muy pequeño de comandos (de uno o dos caracteres de largo cada uno) algunos de los cuales se acompaña con los parámetros numéricos. Por ejemplo, el comando de string "pr,3,20,1" pinta al rango de valores de celda de 3 a 20 con el color especificado por el índice colorimétrico número 1. Los números del índice y los valores correspondientes de color (R, G, B, y valor de la Transparencia) se definen en un archivo de texto que para el acceso del guión deben leerse de un arreglo usando el comando "b."



Se incluyen comandos para crear un archivo de texto de una paleta de colores de en un Archivo de Proyecto, o para crear una paleta de color desde un archivo de texto.

Aunque una interface gráfica es fácil aprender, los usuarios experimentados pueden ejecutar tareas repetitivas más rápidamente usando una interface de línea de comandos. Las tareas que pueden requerir varias acciones del mouse en una ventana gráfica pueden ser ejecutadas usando un solo comando de texto corto.

Más acerca del guión Command Parser esta disponible en el documento en línea en

http://www.microimages.com/relnotes/v65/fixcolor.pdf

Software Avanzado para Análisis Geoespacial

GUIONES SML

MicroImages, Inc. produce una línea completa de software profesional para visualización, análisis y publicación de datos geoespaciales. Contáctenos o viste nuestra página en Internet para información detallada del producto.

TNTmips TNTmips es un sistema profesional con una completa integración GIS, análisis de imágenes, CAD, TIN, cartografía de escritorio y gestión de Bases de Datos geoespaciales.

TNTedit TNTedit provee de herramientas interactivas para crear, georeferenciar y editar materiales de proyectos tipo vector, imagen, CAD, TIN y Bases de Datos Relacionales en una gran variedad de formatos.

TNTview TNTview tiene las mismas características poderosas de despliegue de TNTmips y es perfecto para aquellos que no necesitan las características de procesamiento técnico y preparación de TNTmips.

TNTatlas TNTatlas permite publicar y distribuir materiales de proyectos en CD-ROM a bajo costo. Los CDs de TNTatlas pueden ser usados en cualquier plataforma popular de computadora.

TNTserver TNTserver permite publicar sus Atlas en TNT en Internet o en su Intranet. Navegue a través de atlas de geodatos con su navegador web y el applet Java TNTclient.

TNTlite TNTlite es una versión libre de TNTmips para estudiantes y profesionales con proyectos pequeños. Usted puede descargar TNTlite del sitio Internet de MicroImages, o puede ordenar TNTlite en CD-ROM con el conjunto actualizado de folletos *Tutoriales*.

Indice

MicroImages, Inc.

11th Floor – Sharp Tower
206 South 13th Street
Lincoln, Nebraska 68508-2010 USA

Voice: (402)477-9554
FAX: (402)477-9559

email: info@microimages.com
Internet: www.microimages.com