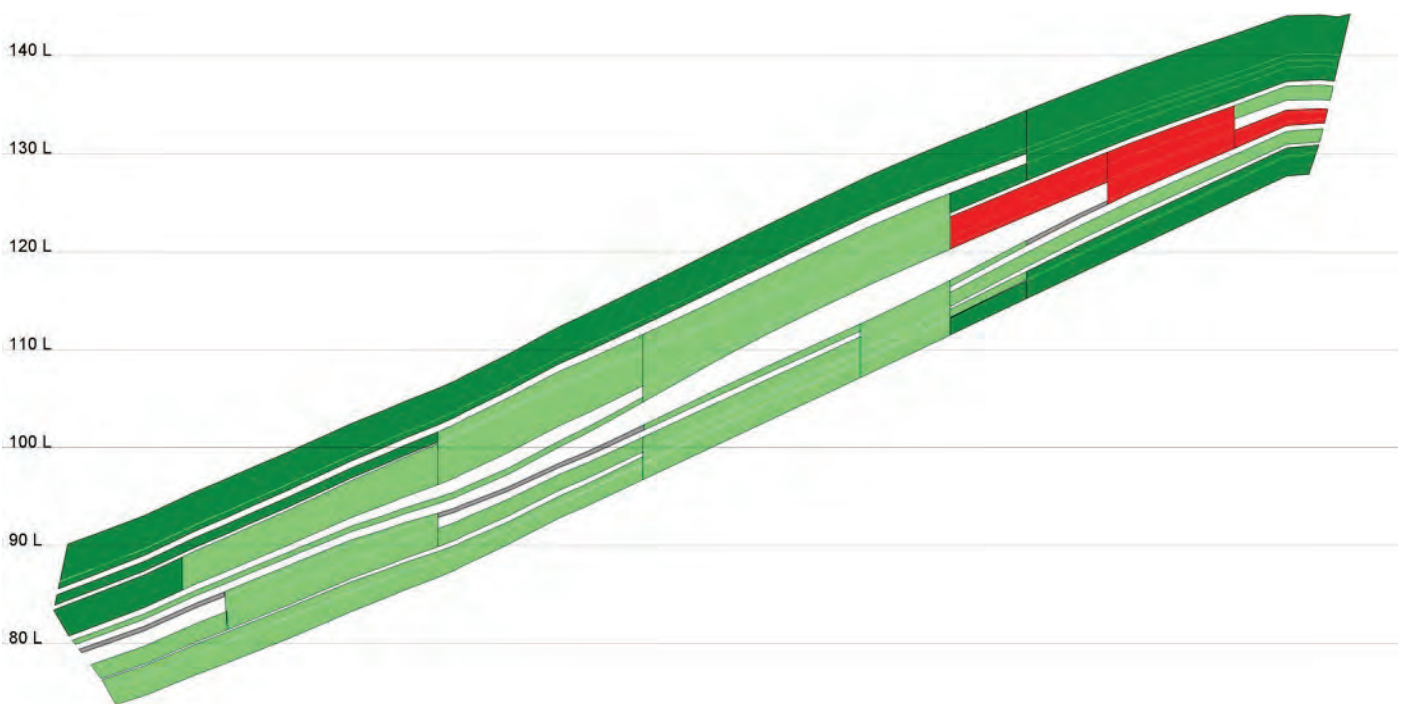




¡LOGRE LO IMPOSIBLE!

Se recomienda el “Scripting” (Secuencia de comandos) como una forma “automática” para ejecutar Vulcan, liberando a los geólogos e ingenieros para que puedan hacer más del “trabajo pensante”.



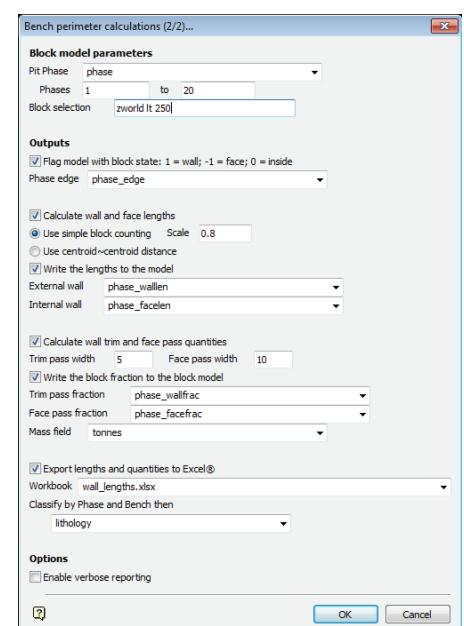
Cada proyecto cuenta con algunas tareas repetitivas, realizadas todos los días, cada hora o muchas veces en cualquier sesión de Maptek Vulcan™. ¿Entonces por qué un script? Se puede usar el scripting para volver a ejecutar comandos frecuentes, simplificando el flujo de trabajo.

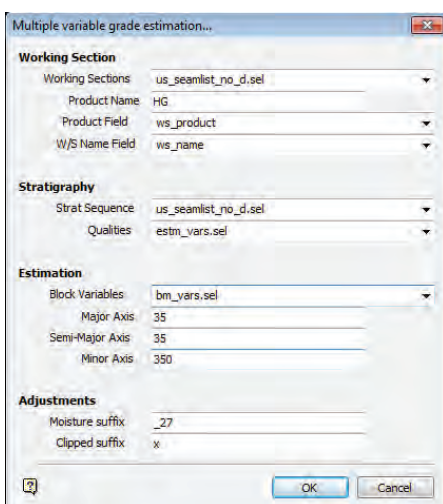
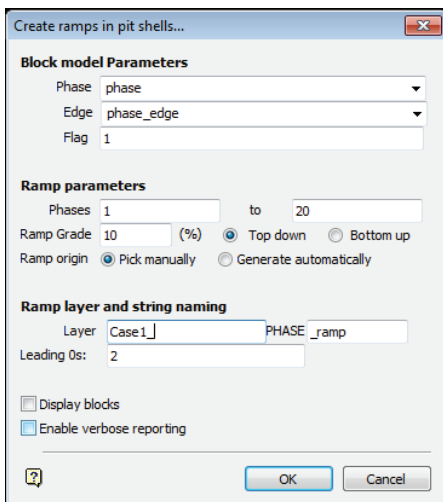
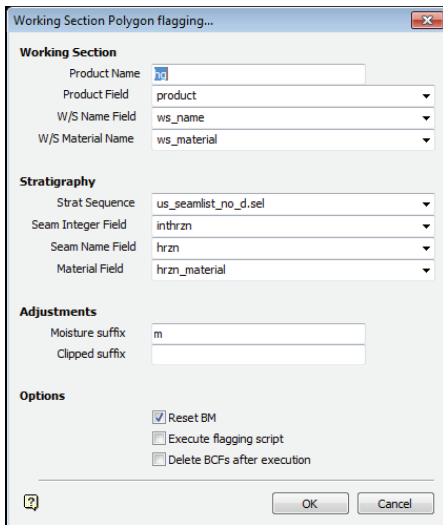
El scripting le permite personalizar o construir procesos. Una tarea de varios pasos complejos puede convertirse en un proceso continuo que se puede ejecutar sin supervisión, ahorrando minutos o incluso horas de su tiempo.

Un beneficio real del scripting Lava es la capacidad de acceder a estructuras de datos tales como triangulaciones, bases de datos, objetos o grillas y manipularlas directamente. Históricamente, los datos se exportaban a ASCII, eran manipulados y luego se volvían a importar.

A los geólogos e ingenieros se les paga para resolver problemas. Crear un registro de auditoría permite la revisión y estandarización de los procesos para múltiples usuarios en múltiples sitios. Bloquear parámetros establece consistencia. Volver a calcular las reservas al final de cada mes es una tarea mecánica. Una vez que haya terminado el proceso, lo puede bloquear en un script y ejecutarlo con seguridad cada mes.

Hay varias opciones para elegir. La función de macro en el menú de archivo es una forma sencilla de automatizar las funciones existentes del menú de Vulcan. Este es un método sencillo para la creación de scripts y para aprender cómo se deben estructurar los mismos.





Las Macros graban las selecciones del menú que hace un usuario. Una vez que entienda cómo funcionan las macros, puede empezar a pasarles variables. Son un muy buen lugar para empezar.

Se puede ejecutar una serie de programas externos a través de la Shell de comandos. Esto es perfecto para los procesos que pueden tardar un par de horas. Inícielos y vuelva más tarde para consultar la respuesta. Puede construir procesos muy complejos, pasando argumentos entre los scripts y los programas, haciendo ciclos iterativos y probando variables. La mayoría de las veces esto se ejecuta en la GUI o mediante la Shell de comandos. Esto requiere más conocimientos de programación que sólo usar la función de las macros.

El scripting Lava o Perl aparece debajo del menú archivo de Vulcan, proporcionando acceso directo a todas las estructuras de datos de Vulcan. Esto le permite controlar la ventana de gráficos y crear sus propios paneles para obtener la aportación del usuario.

Se pueden ejecutar los scripts Lava desde el menú archivo o se pueden ejecutar de manera externa. Se puede utilizar Lava para reunir a todas las funciones existentes en Vulcan y además construir procesos personalizados. Los conocimientos de programación avanzada no son tan difíciles de aprender.

La mayoría del scripting será una combinación de tres opciones. La función de las Macros le permite ejecutar todas las funciones existentes del menú. El scripting T-Shell puede reunir todos los programas externos y también puede ejecutar scripts Lava desde el script T-Shell. Del mismo modo, puede usar el poder de Perl o Lava para ejecutar programas externos.

Desarrollar una corrida de un modelo de bloques de secciones de trabajo de una mina para un yacimiento de carbón de múltiples mantos es un ejemplo en donde los scripts son útiles.

Un script T-Shell toma argumentos de la línea de comandos para ejecutar una serie de scripts Perl que lean los parámetros de un archivo específico, construir varios scripts GDCalc y ejecutar la versión GDCalc de la línea de comandos. El script comprueba y ejecuta todos los archivos.

La salida incluye grillas in-situ y ROM para cada manto y combinación potencial de secciones de trabajo, plantea polígonos que describen la extensión de cada sección de trabajo y un archivo de registro para comprobar la ejecución exitosa de GDCalc. Los resultados se pueden visualizar en esta etapa.

El segundo paso es marcar las secciones de trabajo en el modelo de bloques. Este script crea triangulaciones sólidas para cada polígono de sección de trabajo, un BCF para marcar el modelo de bloques dentro del sólido respectivo para cada sección de trabajo, un script T-Shell para ejecutar el marcado del modelo de bloques, y posteriormente ejecuta el marcado.

El tercer script crea archivos de estimación de bloques y mapfiles de las grillas de las secciones de trabajo ROM y un script de T-Shell para ejecutar las estimaciones y luego corre las estimaciones.

En resumen, el scripting (secuencia de comandos) puede crear funciones personalizadas y realizar tareas que no están disponibles directamente en las herramientas actuales de Vulcan.

*Agradecimientos a Andrew Vidale
Principal, AVCS*

Presentado en AUSTRALIA 2012

