



Redes con VirtualBox

REDES ENCAJONADAS

Nos acercamos este mes a la herramienta de virtualización VirtualBox, un entorno sencillo y fácil para versiones virtuales de Linux, Unix y Windows. **POR TIM SCHÜRMANN**

VirtualBox está ganando popularidad como alternativa sencilla y efectiva de virtualización. Esta inteligente solución se adapta perfectamente al escritorio normal de Linux, soportando distintas variedades de sistemas invitados Windows, Linux y Unix. La empresa alemana Innotek [1] desarrolla VirtualBox, que se presenta en versiones tanto comercial como bajo software libre (GPL).

El 12 de febrero de 2008, Sun Microsystems anunció un acuerdo para comprar Innotek con el objetivo de integrar el entorno de VirtualBox con las herramientas de desarrollo propias de Sun. Las consecuencias a largo plazo de este acuerdo aún no se conocen, pero en el momento de escribir este artículo, parece que Sun pla-

nea mantener el soporte y seguir con ambas versiones, la cerrada y la software libre. En este artículo vamos a describir cómo iniciarse en la virtualización con VirtualBox.

¿Software Libre o Comercial?

Las versiones comercial y libre de VirtualBox se diferencian en el soporte de dispositivos USB y las funcionalidades de escritorio remoto, que Innotek reserva para la variante de código cerrado. Ambas versiones soportan varios sistemas operativos, como Linux, OS/2 y Windows/DOS, y ambas disponen de interfaz gráfica de administración.

Desde el punto de vista de la instalación, la versión software libre bajo

Linux es un poco más complicada (véase el cuadro “Instalar la Variante Software Libre”). Los paquetes específicos para las distribuciones están mejor integrados.

La distribución que utilicemos probablemente ofrecerá una versión de VirtualBox a través del sistema de administración de paquetes. Si tenemos instaladas las fuentes de instalación correctas, podemos teclear simplemente `apt-get install virtualbox` en Ubuntu, o `zypper in virtualbox` bajo SUSE para comenzar la instalación.

Al pulsar el icono o teclear el comando `Virtual-Box` se inicia la interfaz gráfica de usuario. El icono *New* nos permite crear una nueva máquina virtual, con un asistente que nos ayuda en el proceso de configuración.

En lugar de una verdadera partición, VirtualBox usa un archivo con una imagen de disco. Tiene sentido que optemos por *dynamic* como tamaño de la imagen. Recordemos que la imagen crecerá conforme añadamos conte-

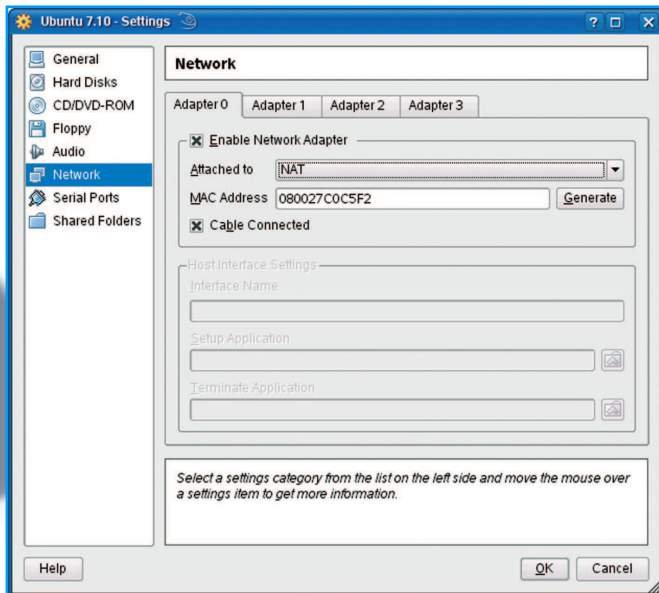


Figura 1: Configuración de la tarjeta de red virtual. El modo NAT se habilita por defecto. El usuario puede generar aquí direcciones MAC e incluso parchear un cable de red virtual para una prueba rápida.

nido. Una vez que la emulación está funcionando, la ventana captura por completo el puntero del ratón. El botón Ctrl derecho funciona como botón del host y libera al ratón.

VirtualBox comprende dos componentes: el demonio XPCOM *VBoxSVC* y el front end *Virtualbox*. Por supuesto, el demonio debe estar ejecutándose antes de iniciar la interfaz gráfica de usuario, pero no es necesario instalarlos de forma separada. Los pasos señalados aquí ejecutarán VirtualBox directamente desde un directorio local.

Conecta Cuatro

Tras completar la instalación, el siguiente paso es seleccionar, en la interfaz gráfica, la máquina virtual cuyas configuraciones de red queremos fijar, cosa que hacemos desde el menú *Machine | Change*. Allí seleccionamos la entrada *Network* en la parte izquierda para acceder a las configuraciones (véase la Figura 1).

Están disponibles hasta cuatro tarjetas de red virtuales para cada máquina, que van desde el *Adapter 0* al *Adapter 3*. Por defecto, sólo se configura un NIC para cada máquina: tendremos que pulsar sobre las pestañas correspondientes para los otros NICs.

El conmutador *Network cable attached* permite al administrador encen-

der o apagar el conmutador de red virtual. La tarjeta se mantiene en la máquina virtual, sin embargo, la conexión al mundo exterior se interrumpe de la misma forma que si alguien hubiese desconectado el cable de red. Este control es útil para hacer pruebas o para reinicializar sistemas invitados con capacidad de conexión en caliente.

Cada una de las cuatro tarjetas de red puede ejecutarse en uno de los tres modos posibles que se especifican en el menú desplegable *Connected to*. Esto especifica con quien se comunica la tarjeta de red virtual y para qué otros orde-

nadores será visible. Las opciones son:

- Red Interna
- Traducción de Direcciones de Red (NAT)
- Modo de Interfaz Anfitrión

Red Interna

VirtualBox es capaz de simular una red LAN interna separada para las tarjetas de redes virtuales, con cualquier número de máquinas virtuales en la red virtual que no verán ni se comunicarán con nadie más, incluidos el sistema anfitrión y los ordenadores fuera de la red interna en la que residen nuestros sistemas invitados (véase la Figura 2).

A primera vista, la red interna no parece particularmente útil, pero hay buenas razones para usarla. Debido a que la LAN virtual está completamente aislada, ninguna influencia externa puede afectarla, lo que hace a la LAN interna una elección perfecta como entorno de pruebas o para resolver problemas.

En los otros dos modos, cualquier información transmitida se envía tam-

Instalar la Variante Software Libre

Para compilar el código fuente, necesitamos las siguientes herramientas y librerías, además de los paquetes de desarrollo:

- GCC
- Ensamblador *as86* (generalmente incluido con el paquete *dev86* o *bin86*)
- Compilador BCC de Bruce Evans (generalmente incluido con el paquete *dev86*)
- Compilador ACPI de Intel (está en las *pmtools* de openSUSE)
- *libxslt*
- *libxerces*
- *libxalan*
- QT versión 3.3.5 o posterior
- *libidl*
- SDL, Alsa y HAL, como el proporcionado por *libhal*

Tras resolver las dependencias, podemos teclear *wget* http://www.virtualbox.org/download/1.5.2/VirtualBox-1.5.2_OSE.tar.bz2 para descargar el archivo con el código fuente [2].

Tras desempaquetar el archivo, tecleamos los siguientes comandos en el

directorio de código fuente de VirtualBox para configurarlo:

```
./configure
source ./env.sh
kmk all
```

Esto instala VirtualBox y su conjunto de herramientas en el subdirectorio *out/linux.x86/release/bin/*. Antes de que podamos iniciarlo, el sistema requiere el correspondiente módulo del kernel:

```
cd out/linux.x86/release/bin/src
make
sudo make install
```

Por supuesto, necesitamos las fuentes del kernel para compilar en este paso. Trabajando como root, podemos habilitar el módulo del kernel tecleando *modprobe vboxdrv* antes de pasar a permitir el acceso a usuarios sin permisos al nuevo archivo de dispositivo: *chmod 666 /dev/vboxdrv*. Ahora ya podemos ejecutar VirtualBox:

```
LD_LIBRARY_PATH=. ./VBoxSVC&
LD_LIBRARY_PATH=. ./Virtualbox
```

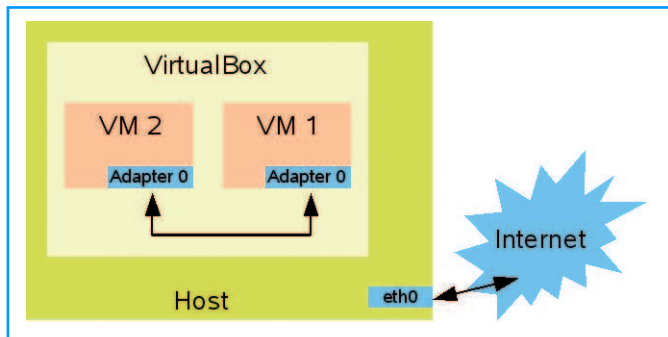


Figura 2: Una red interna protege a las máquinas virtuales frente a los peligros de Internet. Las actualizaciones online no están soportadas, por tanto, en este modo.

bién a la interfaz de red del sistema anfitrión. Si dos máquinas virtuales necesitan intercambiar información, este método casi siempre afecta a la velocidad o presenta problemas de seguridad. Para conectar una tarjeta de red virtual a la LAN interna, seleccionamos *Internal network* bajo *Connected to* en la interfaz gráfica de usuario.

Lo que VirtualBox no te Cuenta

Las tarjetas configuradas de esta manera en cualquier máquina virtual se conectan a un Switch simulado en la misma red virtual. Desafortunadamente, no se dispone de un servidor DHCP en este modo, lo que significa que tenemos que asignar direcciones IP de manera manual. A pesar de que es bastante fácil configurarlas, la interfaz gráfica no nos permite acceder a una de las funcionalidades más interesantes de VirtualBox: el soporte para redes internas independientes, que puede funcionar en paralelo.

LANs Virtuales

Para facilitar la identificación de las LANs virtuales, se asigna un nombre único para cada red. Por defecto, las tarjetas de red que configuremos mediante la interfaz gráfica se conectarán a la red interna llamada *intnet*.

VBoxManage

Si necesitamos más LANs, no tendremos más alternativa que acudir a la línea de comandos VBoxManage. Esta herramienta nos ofrece una alternativa completa en modo texto a la interfaz gráfica de usuario así como la posibilidad de una configuración más

precisa. Sin embargo, su uso requiere teclear largas y crípticas cadenas de parámetros.

Para instalar una segunda interfaz (*nic2*) en una máquina virtual denominada *UbuntuVM* en una LAN interna llamada *MyNetwork*, el *modifyvm* usa la siguiente línea de comandos para modificar la configuración:

```
VboxManage modifyvm >
"UbuntuVM" -nic2 intnet >
-intnet2 "MyNetwork"
```

La red interna es como una caja cerrada: sin una segunda interfaz el sistema invitado es incapaz de navegar por Internet o acceder a la LAN física.

Traducción de Direcciones de Red

El modo NAT nos ofrece un método sencillo y rápido para escapar de esa jaula. Tras seleccionar el modo NAT en el cuadro desplegable *Connected to*, a la tarjeta de red virtual se le asigna una dirección mediante un servidor DHCP simulado con un cortafuegos que da acceso a la máquina virtual al mundo exterior.

La Figura 3 nos muestra cómo funciona esto. En el paso primero, al adaptador de la máquina virtual se le asigna automáticamente una dirección

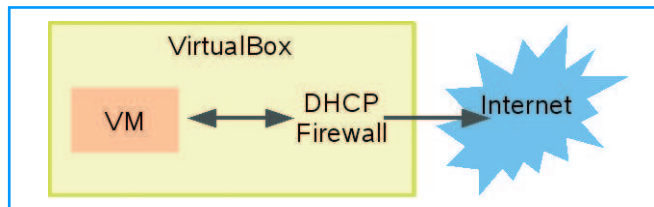


Figura 3: En modo NAT, a la máquina virtual se le asigna una dirección mediante el servidor DHCP interno. El cortafuegos evita que las direcciones externas accedan a la máquina virtual.

IP mediante el servidor DHCP integrado, que normalmente asigna direcciones del rango 10.0.x.x.

Tan pronto como el sistema invitado comienza a transmitir paquetes sobre la conexión virtual, VirtualBox captura estos paquetes e inserta la dirección IP del sistema anfitrión antes de enviarlos a Internet.

A pesar de que la configuración no lleva mucho tiempo, este modo implica una desventaja funcional: la implementación de direcciones IP (NAT) en combinación con el cortafuegos interno permite al sistema invitado enviar información al mundo exterior, sin embargo, el mundo exterior y el anfitrión no pueden acceder al sistema virtualizado. La única excepción a esto es el escritorio remoto de la versión comercial, o bien usar herramientas VPN, como *Openvpn*, para hacer un túnel hasta el interior.

Redireccionamiento de Puertos

En otras palabras, es imposible encapsular servidores en producción en la máquina NAT virtual de manera que sea útil. El redireccionamiento de puertos nos permite solucionar esto. VirtualBox escucha en un puerto del sistema anfitrión y redirecciona los paquetes que llegan a este puerto a otro puerto de la máquina virtual seleccionada. Desde el punto de vista de otra máquina, da la impresión de

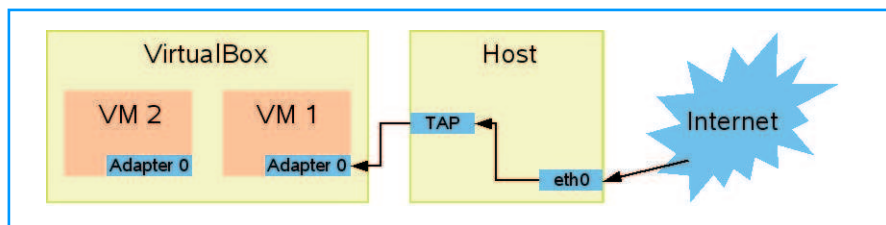


Figura 4: La petición de Internet alcanza al sistema anfitrión, que usa una interfaz TAP para pasarla a la máquina virtual.

Listado 1: Habilitar una Interfaz TAP

```
01 #!/bin/bash
02 # Crear interfaz TAP para
    usuario klaus:
03 interface=`VBoxTunctl -b -u
    klaus`
04 # Crear el interfaz:
05 ifconfig $interface up
06 # Conectar con bridge:
07 brctl addif br0 $interface
```

que el servicio está disponible directamente por el anfitrión.

Son tres los comandos de VBoxManage que permiten el redireccionamiento de puertos antes de iniciar la máquina virtual:

```
VBoxManage setextradata /
"UbuntuVM" "VBoxInternal/
Devices/pcnet/
0/LUN#0/Config/myservice/
Protocol" TCP
VBoxManage setextradata /
"UbuntuVM" "VBoxInternal/
Devices/pcnet/0/LUN#0/
Config/myservice/GuestPort" 22
VBoxManage setextradata /
"UbuntuVM" "VBoxInternal/
Devices/pcnet/0/
LUN#0/Config/myservice/
HostPort" 2222
```

Con esto le indicamos a VirtualBox que redireccione todas las conexiones TCP desde el puerto 2222 del sistema anfitrión al puerto 22 del sistema invitado. *myservice* es un nombre asignable libremente, mientras que *UbuntuVM* es el nombre de la máquina virtual. Si todos estos valores están vacíos cuando se llama al comando, VirtualBox deshabilita de nuevo el redireccionamiento de puertos.

Interfaz Anfitrión

Hacer NAT con redireccionamiento de puertos supone una carga extra notable. El administrador debe abrir, y posiblemente asegurar, puertos en el servidor. Por supuesto, tenemos que llevar la cuenta de qué puertos están mapeados a qué servicios de qué máquina. El modo de Interfaz

Anfitrión nos proporciona la manera más adecuada de ejecutar un servidor en VirtualBox, aunque configurar este modo es un poco más complicado.

En el modo de Interfaz Anfitrión, VirtualBox crea una tarjeta de red virtual adicional en el sistema anfitrión, por ejemplo *vbox0*, además del familiar *eth0*. Tras configurar la nueva tarjeta de red, VirtualBox usa un cable virtual para parchear la conexión al adaptador simulado del sistema anfitrión (véase la Figura 4).

Para aplicaciones en producción, VirtualBox generalmente usa un puente para conectar las redes física y virtual. Esta unión funciona como una única gran red que interacciona con el mundo exterior.

Para configurar esto, en primer lugar necesitamos configurar una interfaz de red virtual para cada uno de los sistemas invitados.

Construir Puentes

Linux tiene adaptadores de redes virtuales en forma de interfaces TAP, y VirtualBox también confía en este método. El único requerimiento es tener acceso libre al archivo de dispositivo */dev/net/tun* con la cuenta del usuario que usa VirtualBox.

De igual manera, necesitamos un software puente, como el paquete *bridge-utils*, y la herramienta en línea de comandos *tunctl*, que es parte del paquete *uml-utilities* en openSUSE.

El puenteo con VirtualBox puede realizarse de dos maneras al comienzo:

- si vamos a tener en todo momento un número de invitados fijo, el método correcto sería configurar una interfaz permanente a la que se conecta la máquina virtual.
- en caso de buscar un método más flexible, podemos permitir que la máquina virtual cree de manera autónoma una interfaz dinámica.

No obstante, tendremos que teclear la contraseña de administración en ambos casos.

Interfaz Permanente o Dinámica

La lista de comandos para configurar una interfaz permanente intimidada bastante.

Lo primero es configurar el puente y conectarlo a *eth0*:

```
brctl addbr br0
ifconfig eth0 0.0.0.0
brctl addif br0 eth0
```

A continuación se actualizan las direcciones IP. Si usamos un servidor DHCP para asignar direcciones, *dhclient br0* se encargará de ello. En caso contrario, necesitaremos asignar direcciones de manera manual con *ifconfig br0 IP address*.

Para cada tarjeta de red simulada en el sistema invitado necesitamos una interfaz virtual del lado anfitrión. *VBoxAddIF vbox0 User br0* se encargará de conectarlo al puente.

El script *VBoxAddIF* crea una interfaz TAP, denominada *vbox0*, y le otorga la cuenta de usuario *User*. *VBoxAddIF* está incluido en el paquete VirtualBox, pero escondido en las profundidades de los directorios del código fuente en *src/VBox/Installer/linux*.

Paso Final

El paso final, trabajando como root, es conectar la tarjeta simulada en el sistema invitado con la interfaz TAP. Para hacerlo necesitamos introducir el nombre de la tarjeta en el cuadro de configuración *Interface name*. En este caso, debemos dejar las dos líneas de scripts en blanco.

Para liberar la interfaz de nuevo tras haberla usado, tecleamos *VBoxDeleteIF vbox0* en la línea de comandos. Por cierto, esta herramienta falta en la variante software libre.

El segundo método supone ejecutar dos scripts: uno para crear la interfaz TAP y conectarla al puente, y el otro para deshabilitar la interfaz.

Listado 2: Deshabilitar una Interfaz TAP

```
01 #!/bin/bash
02 # desconectar el interfaz (el
    nombre está en $2) del
    bridge:
03 brctl delif br0 $2
04 # y eliminar interfaz
05 VBoxTunctl -d $2
```

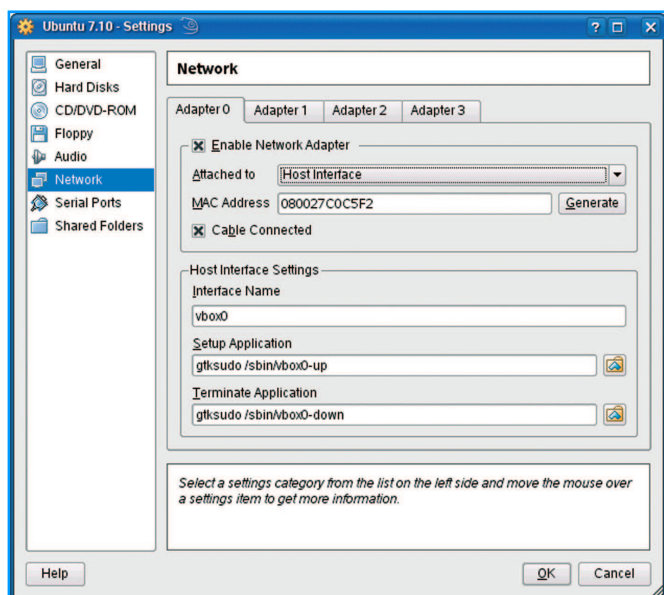


Figura 5: La variante puenteadora Interfaz Anfitrión VirtualBox proporciona al usuario dos cuadros de entrada para iniciar y parar la interfaz virtual. Los scripts parece que funcionan bien aquí

El Listado 1 muestra un script de ejemplo basado en uno proporcionado en el manual para habilitar una interfaz TAP. El manual de VirtualBox [3] proporciona información más detallada acerca de esto.

Herramienta VBoxTuntl

El herramienta *VBoxTuntl* está incluida con VirtualBox: básicamente se trata de la conocida herramienta *tuntl*.

Hacemos ejecutable el script de inicio y lo introducimos en el *Enabling Program* de la interfaz gráfica de configuración de VirtualBox, usando para ello una combinación sudo como *gtksudo startscript* (véase la Figura 5).

Si no disponemos de la interfaz gráfica, podemos hacer esto en su lugar:

```
VBoxManage >
modifyvm >
"VMName" >
-tapsetup1 >
"gtksudo >
startscript"
```

El Listado 2 muestra el script para deshabilitar la interfaz TAP. De nuevo, debemos introducir la ruta hasta el script ejecutable en el campo de entrada de VirtualBox, preferiblemente en combinación con un comando sudo en *Disabling Program*.

En este caso, la alternativa en línea de comandos usa *VBoxManage* con la opción *-taptermiate1*.

Una vez que la máquina virtual ha arrancado, llama al script que habilita para crear una interfaz TAP y nos dice la contraseña de root debido a la llamada a *gtksudo*. Cuando se apaga el sistema invitado, el segundo script elimina la interfaz TAP.

Escritorio Remoto

La variante comercial de VirtualBox es capaz de redirigir la salida de pantalla hasta otro ordenador mediante el Remote Desktop Protocol (RDP), permitiendo de esta manera que los usuarios accedan al escritorio de la máquina virtual directamente desde la red, sin importar la configuración de red que hayamos elegido para la máquina virtual. Esto hace que VirtualBox sea una alternativa para servidores sin interfaz gráfica.

Para habilitar lo que el fabricante denomina Remote Display, en primer lugar tenemos que especificar el número de puerto que VirtualBox va a usar para ofrecer el servicio en el sistema anfitrión. Para evitar que el escritorio de la máquina virtual esté abierto al público, también tiene sentido especificar un método de autenticación.

External significa que el usuario debe registrarse en el sistema anfitrión a la manera estándar de éste. *Guest* significa loguearse en el sistema invitado. Por último, *Null* deshabilita completamente la autenticación.

Tras guardar la nueva configuración, reiniciamos la máquina virtual en línea de comandos tecleando *VBoxVRDP -startvm*. El usuario puede iniciar entonces *rdesktop* o cualquier otro cliente RDP para

visualizar el escritorio de la máquina virtual en sus puestos de trabajo.

Difícil, Estable y Rápido

Los tres tipos de conexión que ofrece VirtualBox para conectar las máquinas virtuales tienen ventajas y desventajas, y la solución óptima dependerá en realidad de las circunstancias.

Si sólo necesitamos acceder a Internet desde la máquina virtual, el modo NAT es una opción útil, pero puede hacer que el acceso remoto sea imposible. En el caso de un servidor en una zona de pruebas, nuestra única opción es la compleja configuración de Interfaz Anfitrión con las herramientas para los puentes.

Lista de Peticiones

Debido a que la interfaz gráfica de usuario actualmente no lo soporta, la implementación de todas las configuraciones de red en la interfaz gráfica es el principal punto que yo añadiría a una lista de peticiones a la empresa Innotek. Dicho esto, los competidores de Innotek no lo han hecho mucho mejor. Configurar las redes en Xen es igualmente complejo, y VMware sólo ofrece una herramienta en línea de comandos denominada *vmware-config.pl* para redes virtuales, aunque la herramienta puede configurar múltiples redes NAT, sólo-host y modo puente en unos pocos pasos.

Conclusiones

La solución podría estar a la vuelta de la esquina. Se rumorea que Innotek está trabajando duro en la configuración de redes. Si prefiere no esperar, será bien recompensado por el esfuerzo que ponga en ello. Una vez que haya configurado VirtualBox de manera adecuada, podrá avanzar enormemente con estos sistemas virtuales rápidos y estables. ■

RECURSOS

- [1] Página de VirtualBox: <http://www.virtualbox.org>
- [2] Edición Open Source: http://www.virtualbox.org/download/1.5.2/VirtualBox-1.5.2_OSE.tar.bz2
- [3] Manual de usuario: http://www.virtualbox.org/wiki/End-ser_documentation