

ExprTk

Manual de instalación y uso

Proyecto Simusol

Octubre de 2026

Índice

1. Qué es ExprTk	1
2. Instalación, documentación y aspectos técnicos	1
2.1. Instalar	1
2.2. Construir y ver la documentación	2
2.3. Técnicos	2
2.3.1. Instalar desde fuentes	2
2.3.2. Construir un paquete binario desde fuentes	2
2.3.3. Instalar desde un paquete construido	2
2.3.4. Dependencias	3
3. Uso y documentación complementaria	3

1. Qué es ExprTk

Este paquete distribuye la cabecera ExprTk y su archivo pkg-config para los programas del proyecto. ExprTk compila expresiones matemáticas en memoria y permite evaluarlas repetidamente con variables diferentes.

2. Instalación, documentación y aspectos técnicos

2.1. Instalar

Para instalar ExprTk mediante SummaPack:

```
curl -fsSL0 https://www.simusol.org/downloads/installsumma.sh  
sh installsumma.sh exprtk
```

Se necesita acceso a Internet y curl. El instalador recibe el proyecto y resuelve sus paquetes y dependencias; solicita permisos cuando debe modificar el sistema. Revisar el script antes de ejecutarlo. Para DEB se puede elegir expresamente **-para deb**; para compilar e instalar en

la máquina, **-para fuentes**. No confundir **summapack exprtk** con el instalador: esa orden no implementa la instalación. La disponibilidad depende del catálogo publicado para el proyecto y de versiones compatibles; no usar opciones que ignoren dependencias para disimular errores.

2.2. Construir y ver la documentación

Desde la raíz del proyecto:

```
make -C doc/manual
xdg-open doc/manual/manual.pdf
```

La última orden abre el visor del escritorio; en un servidor sin entorno gráfico basta con construir y descargar el PDF. Se requiere LaTeX con español y los paquetes usados por el documento; en Debian/Ubuntu, **latexmk**, **texlive-latex-extra**, **texlive-lang-spanish** y **lmodern** proporcionan el entorno habitual. Las fuentes se conservan; el PDF y los auxiliares se regeneran. No se necesita LaTeX para ejecutar ExprTk. Construir en una copia escribible, no bajo el directorio de documentación instalado por el sistema.

2.3. Técnicos

2.3.1. Instalar desde fuentes

Desde una copia de las fuentes, con las dependencias de desarrollo disponibles:

```
make PREFIX="$HOME/.local"
make install PREFIX="$HOME/.local"
```

Este ejemplo usa un prefijo privado. Añadir su **bin** a **PATH** si se instalan ejecutables. Si se usa el prefijo del sistema, reservar **sudo** para la instalación, no para compilar o probar. La instalación desde fuentes y la instalación mediante el gestor de paquetes son recorridos diferentes; no mezclar versiones sin controlar qué archivos y bibliotecas se usan.

No hay configure ni biblioteca compartida de ExprTk. El Makefile genera **exprtk.pc** e instala la cabecera desde la lista de **exprtk.summa**; **make uninstall** usa esa misma lista. No ejecutar los makefiles de experimentos históricos en varios/ como si fueran el constructor actual.

2.3.2. Construir un paquete binario desde fuentes

Desde la raíz del proyecto:

```
./hacer-pkg
```

El manifiesto **exprtk.summa** describe los paquetes; SummaPack genera los metadatos de empaquetado en una copia de construcción. El constructor utiliza las herramientas de Debian para preparar DEB y deriva TGZ/RPM cuando están disponibles sus herramientas; no instala ni publica por el hecho de construir. Necesita SummaPack y los paquetes de desarrollo que declare **BUILDDEPENDS**. Las bibliotecas usadas para empaquetar deben poder identificarse con sus paquetes, no solo existir como copias sueltas en un prefijo privado. Los artefactos y sus sumas de comprobación se registran en la lista de publicación. No sobrescribir una versión ya publicada para introducir cambios nuevos.

2.3.3. Instalar desde un paquete construido

Ejemplo DEB, desde el directorio que contiene los artefactos:

```
sudo apt install ./libexprtk-dev_2025.01.01-1_all.deb
```

Sustituir los nombres por los archivos exactos de la versión construida, sin seleccionar varias versiones mediante un comodín. APT descarga lo que conoce en sus repositorios; las dependencias de Simusol se obtienen mediante SummaPack o se agregan a la misma orden como paquetes locales compatibles. `dpkg -i` por sí solo no descarga dependencias. No forzar su omisión.

Para RPM usar el gestor de la distribución, como `dnf install` o `zypper install`, con las rutas exactas. Para un TGZ del proyecto usar `instalar-tgz`, consultando primero su ayuda; no extraerlo indiscriminadamente sobre la raíz del sistema. El formato no garantiza compatibilidad ABI con cualquier distribución. Para desinstalar un binario, usar el mismo gestor que lo instaló.

2.3.4. Dependencias

Del proyecto Simusol. El instalador de fuentes usa `instalar-del-summa` de SummaPack. La cabecera no enlaza bibliotecas de Simusol. `Calcula` y `Tabular2` son consumidores; no se requieren para instalar `libexprtk-dev`.

De software externo. `ExprTk`, de Arash Partow, es una cabecera C++ con licencia MIT. Los programas que la incluyen necesitan un compilador C++; instalar la cabecera y su archivo `pkg-config` no requiere compilar una biblioteca.

De lenguajes y módulos. No usa módulos Perl ni Python para evaluar expresiones. Las herramientas de empaquetado tienen sus propias dependencias; no son requisitos del programa C++ consumidor.

De documentación. LaTeX y sus paquetes; el visor es opcional. Estas herramientas no son dependencias de ejecución.

`DEPENDS` y `BUILDDEPENDS` del manifiesto distinguen ejecución y construcción. Los paquetes de bibliotecas agregan dependencias transitivas y los binarios nativos declaran los mínimos determinados al enlazarse. Los nombres pueden cambiar entre distribuciones: SummaPack mantiene equivalencias; un nombre de proyecto, un paquete binario y un ejecutable no siempre coinciden.

3. Uso y documentación complementaria

```
pkg-config --cflags exprtk
```

El consumidor incluye `exprtk.hpp`, registra variables en una tabla de símbolos y compila su expresión. Este paquete no instala una terminal llamada `exprtk`. El soporte de funciones propias, como tablas o ecuaciones físicas, corresponde a quien incorpora la cabecera. Las fuentes de referencia están en `src/packages/exprtk`; los experimentos de `varios/` no forman parte de la interfaz instalada.