Comunicaciones Unificadas con Elastix

Volumen 2

2da Edición

Edgar Landivar

Comunicaciones unificadas con Elastix es una obra destinada a los usuarios y entusiastas de esta distribución. Sirve también como material de referencia para aquellos que quieren entrar en el mundo de Voz sobre IP.

Desde su primera edición, este libro ha sido material de capacitación para la certificación oficial de Elastix y también ha servido como punto de partida en la carrera profesional de muchos integradores que eligieron Elastix como principal solución de las Comunicaciones Unificadas.

En un lenguaje simple, el autor lleva al lector hacia la implementación de configuraciones complejas. Recomendado para aquellos que desean potenciar su conocimiento en telecomunicaciones.

Segunda Edición

A Danielita, por su sonrisa…

Reconocimiento

La elaboración de este libro involucró un enorme trabajo que fue facilitado gracias a la generosa colaboración de algunas personas con las cuales me encuentro enormemente agradecido. Mis agradecimientos más especiales a:

- *José Landívar* y *Alfredo Salas* de ElastixDepot LLC, por la prolija revisión del volumen 1, párrafo a párrafo.
- A mi *Esposa*, por la comprensión durante el tiempo que le tuve que dedicar a la escritura del presente libro.
- A *Paul Estrella*, por ejercer la presión necesaria para que esta obra se termine a tiempo.
- A *Bob Fryer* por su importante ayuda revisando la versión en Inglés.

El libro *online*

A partir de esta segunda edición el libro contará con su propio sitio Web en http://www.elastixbook.com. Desde aquí el lector podrá acceder al libro en formato HTML.

¿Dónde adquirir este libro?

La versión impresa, tanto como la versión digital, pueden ser adquiridas a través de Lulu.com. Un hipervínculo está disponible en http://www.elastixbook.com.

Feedback

Como siempre, cualquier sugerencia será bien recibida. Que el lector no dude en escribir un email a elastix-book@palosanto.com

17

Plan de marcado avanzado

Olvidar lo malo también es tener memoria
-- José Hernandez en su obra -Martín Fierro-

17.1 Introducción

Aunque la mayoría de configuraciones telefónicas en Elastix pueden realizarse a través de la interfaz Web y no es necesario tocar los archivos de texto donde reside el plan de marcado, a nivel del ingeniero de soporte es necesario conocer las entrañas de nuestro sistema para realizar configuraciones complejas.

En el volumen 1 exploramos brevemente el plan de marcado (*dialplan*) de Asterisk explicando conceptos básicos como contextos, aplicaciones y variables. En este capítulo avanzaremos más y explicaremos conceptos adicionales

que nos permitirán leer y escribir plan de marcado con más facilidad.

17.2 Manipulación de variables

Eliminación de dígitos al inicio y final de una variable

En ocasiones necesitaremos modificar una variable como ${EXTEN} y eliminar algunos dígitos al principio del número ya que estos muchas veces son prefijos telefónicos. Por ejemplo, si queremos marcar a través de un proveedor VoIP que termina llamadas exclusivamente en USA puede ser que sea necesario que el número telefónico deba ser enviado sin el prefijo 001 con el que fue marcado desde un teléfono interno.

Este tipo de cosas se logran de manera sencilla usando la sintaxis apropiada que nos permite eliminar estos dígitos iniciales.

La sintaxis usada para eliminar dígitos al inicio y/o al final de una variable es **${NOMBRE_VARIABLE:x:y}**

Donde -x- representa el número de dígitos que se eliminarán al principio de la variable y -y- representa el número de dígitos que se extraerán de dicha variable.
Por ejemplo, supongamos que originalmente la variable ${EXTEN} contiene el número telefónico 123456789. Por lo tanto la expresión ${EXTEN:2:4} nos devolverá el número

3456. Es decir que se eliminaron **2** (x) dígitos al principio y el número devuelto debe contener **4** (y) dígitos.

Si se omite -y- en la expresión (lo cual es el caso más común) entonces se asume que se devolverán todos los restantes dígitos contenidos en la variable. En el ejemplo del número telefónico anterior con la expresión ${EXTEN:2} obtendríamos 3456789.

Concatenación de variables

Para concatenar dos o mas variables simplemente debemos escribirlas juntas. Por ejemplo, para concatenar las variables VAR1, VAR2 y VAR3 podríamos hacer algo como lo siguiente.

```
exten => s,n,SetVar(VAR_CONCAT=${VAR1}${VAR2}${VAR3})
```

Expresiones

Una expresión es una combinación de variables y operadores que producen un resultado. Un uso común de una expresión es el de realizar operaciones matemáticas como por ejemplo sumar dos valores. Una expresión tiene la siguiente sintaxis.

$[expresión]

Por ejemplo, para sumar dos variables llamadas VAR1 Y VAR2 podemos utilizar la expresión $[${VAR1} + ${VAR2}]

Operadores

Existen operadores de diferentes tipos como *booleanos*, matemáticos, de comparación y para manipulación de expresiones regulares.

Operador	Descripción
Booleanos	
expr1 | expr2	Operador OR. Retorna expr1 en caso de que expr1 contenga una cadena no-vacía o diferente de cero. Caso contrario retorna expr2.
expr1 & expr2	Operador AND. Retorna expr1 siempre y cuando tanto expr1 como expr2 contengan cadenas no-vacías o diferentes de cero. Caso contrario retorna cero.
!expr	Operador de complemento.
Para comparar	
expr1 = expr2	Devuelve 1 si la comparación es cierta y 0 si es falsa
expr1 != expr2	Devuelve 1 si la comparación es cierta y 0 si es falsa
expr1 < expr2	Devuelve 1 si la comparación es cierta y 0 si es falsa
expr1 > expr2	Devuelve 1 si la comparación es cierta y 0 si es falsa
expr1 <= expr2	Devuelve 1 si la comparación es cierta y 0 si es falsa
expr1 >= expr2	Devuelve 1 si la comparación es cierta y 0 si es falsa

Matemáticos	
expr1 + expr2	Suma dos valores
expr1 - expr2	Resta dos valores
- expr	Niega un valor
expr1 * expr2	Multiplica dos valores
expr1 / expr2	Divide un valor para otro
expr1 % expr2	Devuelve el módulo de un valor para otro
Para manipular expresiones regulares	
expr1 : regexp	Compara expr1 contra una expresión regular contenida en regexp. La comparación asume un carácter de inicio de cadena ^ implícito al inicio de regexp.
expr1 =~ expr2	Similar al operador anterior excepto que no se asume un carácter de inicio de cadena implícito.

Sintaxis condicional

Cuando se necesita evaluar una expresión y tomar una decisión dependiendo del resultado de dicha evaluación, podemos usar lo que se llama sintaxis condicional y que mostraremos a continuación.

expresión?destino1:destino2

Es decir que si la expresión es verdadera se retorna destino1 y si es falsa se retorna destino2.

Esta sintaxis es la usada por aplicaciones como GotoIf.

17.3 Macros

Las macros no son otra cosa que aplicaciones definidas por el usuario. Una de las grandes ventajas de esto es que nos permiten reutilizar código de plan de marcado.

Para definir una macro simplemente escribimos el código de plan de marcado como si fuera un contexto cualquiera con el cuidado de que el nombre de este contexto debe empezar con la palabra -macro- seguida de un guión y del nombre de nuestra macro. Por ejemplo:

```
[macro-nohacenada]
exten => s,1,NoOp(No se hace nada)
```

Para invocar la macro lo hacemos a través de una aplicación especial llamada también Macro, la cual tiene la siguiente sintaxis.

```
Macro(nombremacro,arg1,arg2…)
```

Como vemos, la aplicación Macro toma como primer parámetro el nombre de la macro y luego los argumentos que la macro pueda tener.

Si queremos que nuestra macro manipule argumentos podemos acceder a ellos automáticamente a través de las variables ${ARG1}, ${ARG2}, ${ARG3}, etc

Por ejemplo, para invocar la macro que creamos antes podemos proceder de la siguiente manera.

```
exten => s,n,Macro(nohacenada)
```

Veamos a continuación la definición de una macro muy usada en el plan de marcado propuesto por FreePBX y que sirve para colgar una llamada.

```
; What to do on hangup.
[macro-hangupcall]
exten => s,1,ResetCDR(w)
exten => s,n,NoCDR()
```

> ***Nota:*** *La macro anterior es incluida en una versión vieja de FreePBX, pero se ha decidido dejarla aquí por razones didácticas. La actual macro-hangupcall es mucho más compleja que la mostrada aquí.*

Variables disponibles en la macro

Aparte de los argumentos del tipo ${ARG*n*} existen otras variables disponibles en el contexto de una macro. Estas variables son las siguientes.

Nombre variable	Descripción
${MACRO_CONTEXT}	El contexto de la extensión desde donde se llamó a la macro
${MACRO_EXTEN}	La extensión desde donde se llamó a la macro