



# **MANUAL DEL PROGRAMADOR DE SUTIL (v2.10)**

Impreso el miércoles, 26 de octubre de 2011 a las 14:20:31.

Este manual consta de 110 páginas, 27487 palabras y 3 imágenes.

Referencia Aplicación: DSL108

## Índice de contenido

1.- Introducción.....	8
2.- Programación de Scripts IVR en SUTIL.....	10
2.1.- Generalidades.....	11
2.1.1.- Cuestiones básicas.....	11
2.1.2.- Llamadas a los ficheros script.....	12
2.1.3.- Recarga de ficheros script.....	12
2.1.4.- Códigos de resultado estándar.....	12
2.1.5.- Comandos información estado Sutil.....	13
2.2.- Directivas de pre-compilación.....	14
2.3.- Comandos de control de ejecución.....	15
2.4.- Gestión de variables.....	21
2.5.- Gestión y conexión de canales.....	29
2.6.- Utilización de recursos de voz.....	49
2.7.- Gestión de extensiones y grupos.....	60
2.8.- Comunicación a través de sockets IP.....	66
2.9.- Gestión de interrupciones.....	70
2.10.- Gestión de buzones vocales.....	73
2.11.- Contadores externos.....	76
2.12.- Generación del histórico de llamadas.....	80
2.13.- Lista negra de teléfonos.....	82
2.14.- Generación de fichero de trazas.....	85
3.- Funciones de Sutil asociadas a un script.....	87
3.1.- Operaciones ejecutables desde un terminal telefónico conectado a una extensión que ejecutan un script.....	88
3.1.1.- Asignar número de extensión lógica. (LOGIN).....	88
3.1.2.- Desasignar número de extensión lógica. (LOGOUT).....	88
3.1.3.- Escuchar los mensajes grabados en nuestro buzón.....	88
3.1.4.- Escuchar los mensajes grabados en otro buzón.....	88
3.1.5.- Grabar mensaje de bienvenida de nuestro buzón.....	88
3.1.6.- Dejar un mensaje a un buzón concreto.....	88
3.1.7.- Pasar llamada a una extensión concreta sin esperar.....	88
3.1.8.- Ejecutar script de usuario.....	89
3.2.- Funciones ejecutables desde el Monitor de Estado de Sutil que ejecutan un script.....	90
3.2.1.- Retener llamada.....	90
3.2.2.- Transferir llamada.....	90

4.- Ficheros Scripts de ejemplo.....	91
4.1.- Como centralita de empresa.....	92
4.1.1.- Todos nuestros operadores están ocupados. Espere unos momentos... 92	
4.1.2.- Se pide una opción y si no se pulsa nada se pasa la llamada al grupo 200.....	93
4.1.3.- Teclee el número de extensión o espere o pulse 0 para hablar con una operadora.....	95
4.1.4.- "Teclee el número de extensión" con opciones de salida.....	97
4.1.5.- Información asociada al número de teléfono del llamante.....	98
4.2.- Empresas de 80x.....	99
4.2.1.- Carrusel general.....	99
4.2.2.- Mailbox general.....	99
4.2.3.- Reproducción de mensaje de TAROT pregrabado.....	100
4.2.4.- Autenticación de operadores externos.....	102
4.2.5.- Pago de visita a web a través de 803/806.....	104
4.2.6.- Dedicar canciones.....	105
4.2.7.- Validar a un usuario y conectarlo a un salón.....	108
4.2.8.- Aplicación de tarjetas pre-pago.....	109

## Índice alfabético de comandos para ficheros Scripts

BEEP Canal.....	54
BL_EXIST NumTelefono NumVar.....	83
BL_INIT.....	82
BL_INSERT NumTelefono NumVar.....	82
BL_REMOVE NumTelefono NumVar.....	83
CALCWAITTIME NVar NExt [Tmedio].....	64
CALL FicheroIVR Arg1 Arg2 .....	17
CALL2EXTENSION Canal NExt Timeout [MensajeLCD].....	30
CALL2EXTENSIONBG Canal NvarEstado Next CanalSecEsc SecEsc Timeout [MensajeLCD].....	31
CALL2EXTERIOR Canal NTelefono Timeout [NLinea1] [NLinea2] .....	33
CALL2EXTERIORBG Canal NvarEstado NvarCausa NTelefono CanalSecEsc SecEsc Timeout [NLinea1] [NLinea2] .....	34
CALL2EXTERIORISDN Canal NTelefono Timeout NvarCausa [NLinea1] [NLinea2] .....	36
CALL2VOIP Canal UsuarioSip@DireccionIP Timeout.....	47
CALLSIMULTANEOUSBG Canal NvarEstado NvarCausa CanalSecEsc SecEsc Modo TimeOut ListaNumerosyCanales.....	37
CHECKVOICE Canal Nvar TiempoSilencio.....	58
CLEARINPUT Canal.....	58

CMP2TIME Time1 Time2 Numvar.....	27
CMPTIME Time Numvar.....	26
CONNECT Canal1 Canal2 [Modo].....	41
CONNECT2ROOM Canal Room [Opcion [NVarRespuesta]].....	42
CONNECT2SPEAKER Canal.....	41
CONVDATE Date NumVar.....	27
CONVTIME Time NumVar.....	27
COUNTER_INC Contador NumVar [NumDigitos].....	77
COUNTER_INIT.....	76
COUNTER_READ Contador NumVar [NumDigitos].....	78
COUNTER_RESET Contador NumVar [ValorInicio].....	76
CREATETMPGROUP NumVar Distribución RetryCall ListaExtensiones.....	64
DECGLOBAL NumVar.....	24
DECLocal NumVar.....	23
DISABLEINT.....	71
DISPLAYLCD NExt MensajeLCD.....	62
ENABLEINT.....	72
EVAL NumVarLoc ExpAlgebraica.....	23
EXIT.....	15
GENCALLREG Canal.....	81
GETINFOCANAL Canal NVarTipo NVarNum NVarDir.....	43
GETINFOGROUP NExt NVarTotal NVarAuten NvarTipo [ NvarActi [ NvarDispo [ NvarCola ] ] ].....	44
GETINFOROOM NumSalon NumVar.....	42
GETINFOSCRIPT Script NumVar.....	13
GETINFOSERVICE Servicio NumVar.....	13
GETNUMCALL Canal NvarLla NvarSeg NvarGra.....	81
GETOPENCHANNEL Canal Tipo Linea.....	40
GOSUB etiqueta.....	15
GOTO etiqueta.....	15
HOOK Canal [Causa].....	41
IF cadena1 comp cadena2 <ELSE> ENDIF.....	16
IFERR Nerr1a [Nerr1b] ... ..Instrucciones... .... ENDIFERR.....	17
INCGLOBAL NumVar.....	25
INCLocal NumVar.....	23
INCLUDE ficheroIVR.....	14
INCOMINGCALL Accion [Causa].....	29
INPUT Canal NVar Ncharacter Timeout [CaracteresFin [CaractAdmitidos]].....	57
INPUTR Canal Timeout.....	58
INTDATE DateFrom DateTo NumVar.....	27
INTTIME TimeFrom TimeTo Numvar.....	27
INVOKE FicheroIVR Arg1 Arg2 .....	19
JUMP FicheroIVR Arg1 Arg2 .....	18

LOCKGLOBAL NVar Timeout.....	24
MACRO Comando Arg1 Arg2 .... ..Instrucciones... ENDM.....	14
MBOXCLOSE.....	73
MBOXCREATE NBuzon NumVar Clave Email.....	73
MBOXEMAIL NBuzon NVar.....	75
MBOXEXIST NBuzon NVar.....	75
MBOXMAX NumVar.....	75
MBOXMENU Canal NBuzon Clave.....	74
MBOXNUMARC NBuzon NVar.....	74
MBOXNUMNEW NBuzon NVar.....	74
MBOXOPEN NGrupo.....	73
MBOXPLAYWM Canal NBuzon.....	75
MBOXRECORDM Canal NBuzon Opcion1 Opcion2.....	74
MBOXRECORDWM Canal NBuzon Clave Opcion.....	74
MBOXREMOVE Nbuzon Clave.....	75
MOD Dividendo Divisor NumVar.....	28
ONHOOK Canal [Etiqueta].....	42
ONSOCKET Handle [ NvarRespuesta Etiqueta ].....	69
OPENFILE Handle Modo Fichero.....	85
OPENSOCKET Handle DireccionIP Puerto.....	68
PAUSE [Tiempo].....	15
PLAYDTMF Canal Cadena.....	53
PLAYFILE Canal Fichero [ Opcion [ NivelTraza ] ].....	49
PLAYMONEY Canal Cadena [Opcion [Moneda [Idioma]]].....	51
PLAYNUMBER Canal Cadena Genero [Opcion [Idioma]].....	52
PLAYORDINAL Canal Cadena Genero [Opcion [Idioma]].....	52
PLAYTIME Canal FormatoFecha FormatoHora Cadena [Opcion [Idioma]].....	50
PLAYTONE Canal [TipoTono [ModoParada]].....	53
RANDOM Desde Hasta Numvar.....	26
READUSERDATA Canal NVar.....	48
RECEIVEFAX Canal NvarRet Fichero.....	45
RECORDFILE Canal Fichero Maxtiempo Opción [TiempoSilencio [Attach]].....	54
RECORDFILEBG Canal Fichero Maxtiempo Opción [TiempoSilencio [Attach]].....	56
RESERVEEXTENSION NExt [NVar Etiqueta [Prioridad [Esperando]]].....	63
RETSUB.....	16
RETURN Valor.....	18
SELECTCASE cadena1 CASE <CASEELSE> ENDSELECTCASE.....	16
SEND2SOCKET Handle NVarRespuesta Timeout CadenaEnvio.....	68
SENDFAX Canal NVarRet Fichero NumFax [NombreDestino].....	46
SENDUSERDATA Canal Texto.....	48
SETCALLRIGHTS Rights.....	61
SETCODECALL CanalCodigo.....	80

---

SETERROR Nerr.....	26
SETEXTENSION NumExt Tipo [Asignación] [Password].....	60
SETGLOBAL NumVar [Cadena].....	24
SETLANGCALL Canal CodIdioma.....	81
SETLOCAL NumVar [Cadena].....	23
SETLOCALIDFAX [Identificación].....	47
SETOFFHOOKTIMEOUT Canal Time.....	40
SETPRIORITYSCRIPT Priority.....	19
SETSELFALLERID [NúmeroPropio [HIDE]].....	39
SETTYPECALL Canal Tipo.....	80
STRSUB NumVarDest Origen Longitud Cadena.....	25
STRTOKEN NumVarDest NumToken Separador Cadena.....	25
STRTOKENMULT NumVarInicial MaxTokens NumVarLeidos Sep Cadena.....	25
TRANSFER2EXTENSION NExt Timeout MensajeVoz [MensajeLCD].....	62
TRANSFER2EXTENSIONASK NExt Timeout [MensajeLCD].....	63
UNCONNECT Canal.....	41
VAR NomVar .....	22
WRITE2FILE Handle Cadena.....	86
WRITE2TRACE Handle NivelError Cadena.....	86

## **1.- Introducción.**

SUTIL es un sistema de telefonía computarizada totalmente autónomo. Hace todas las funciones de una centralita telefónica de cualquier tipo, por lo que puede utilizarse en cualquier entorno donde podría funcionar una centralita.

Pero, además de ello, SUTIL puede realizar otras muchísimas funciones. Prácticamente cualquier labor que tenga que ver con automatizar procesos telefónicos.

SUTIL se puede aplicar principalmente en estos 7 entornos:

- Como centralita de empresa. Añade innumerables ventajas: grabar conversaciones, operadora virtual, etc.
- En entornos de telefonía de coste añadido 80x. Es la plataforma ideal en estos entornos.
- En contact-centers y call-centers. La mejor solución tecnológica para este tipo de entornos.
- En hoteles, residencias, hospitales, etc. Es la plataforma ideal para estos entornos, permite buzones para cada habitación, gestión inteligente del centro a través del teléfono, etc.
- Como sistema de enrutamiento masivo de llamadas: Gestión de tarjetas pre-pagos, pasarelas de llamadas, etc.
- En Centros de negocio. Permite una gestión inteligente de las empresas albergadas.
- Como plataforma de respuesta automatizada IVR. Funcionando de forma independiente o conectado a una centralita mayor puede proporcionar información de bases de datos a cualquier llamante.

Sutil permite definir Extensiones Lógicas, de forma que pueden autenticarse en una extensión física o en un número de teléfono. Esto permite que un operador/a trabaje un día en su casa, otro día trabaje en una extensión física y otro día en otra extensión física, pero que en cualquier caso podemos obtener información de toda la actividad de esa extensión lógica, independientemente de que haya estado autenticada en distintos sitios.

Sutil también nos permite asignar un scripts a una extensión lógica, de forma que al llamar a esa extensión lógica nos respondera ese script.

De esta forma cuando llamamos a una extensión lógica, Sutil, según proceda, llama a una extensión física, si es necesario llama a un número de teléfono o nos responde un script.

Sutil define las llamadas como Entrantes, Salientes o Internas, tanto a bajo nivel como a alto nivel.



Como bajo nivel entendemos el nivel físico, es decir cuando sutil llama a un número de teléfono es una llamada Saliente, cuando Sutil recibe una llamada de teléfono es una llamada entrante y cuando desde una extensión física se llama a otra extensión física es una llamada interna.

Como alto nivel entendemos el nivel lógico, es decir cuando Sutil llama a una extensión que esta autenticada en un número de teléfono, esa llamada a alto nivel es interna, ya que se trata de una llamada a una extensión, sin embargo a bajo nivel es una llamada saliente.

Igualmente si Sutil recibe una llamada desde un número de teléfono que esta asignado a una extensión lógica, a alto nivel se considera una llamada interna, ya que proviene de una extensión, y a bajo nivel es una llamada entrante.

Sutil nos permite listar las llamadas filtrandolas por entrantes, salientes o internas tanto a bajo nivel como a alto nivel. De esta forma podemos, por ejemplo, listar todas las llamadas salientes a bajo nivel para comprobar la factura de nuestro operadora de telefonía, o listar todas las llamadas salientes a alto nivel a una extensión para comprobar todas las veces que hemos llamado a esa extensión lógica.

## ***2.- Programación de Scripts IVR en SUTIL.***

Una de las claves más importantes de la potencia de SUTIL está en su sistema de scripts. Cuando SUTIL recibe una llamada, dependiendo de a que número ha llamado el cliente se selecciona que se ejecute un determinado script. Igualmente se puede hacer que se ejecute un script asociado a llamadas salientes.

Mediante un script podemos programar un completo sistema IVR que interactúe con el cliente, una operadora virtual, sistemas de suministro de información a clientes, sistemas de encaminamiento masivo de llamadas, etc.

Esto hace de SUTIL un sistema muy versátil ya que en pocas horas se puede configurar para que realice todo tipo de tareas diferentes. La programación de scripts es sencilla y está al alcance de cualquiera aunque jamás haya realizado un programa.

## 2.1.- Generalidades.

### 2.1.1.- Cuestiones básicas.

Los ficheros tienen extensión **.ivr** y tienen que estar grabados en el disco duro de SUTIL dentro del directorio **Scripts** dentro del directorio de SUTIL (normalmente **C:\Sutil**). El nombre del fichero puede tener como máximo 20 caracteres más la extensión, en el fichero histórico solo se almacenan como máximo los 20 primeros caracteres del nombre del script incluyendo la extensión.

La sintaxis de los comandos no distingue entre mayúsculas y minúsculas. Los comandos pueden ir seguidos de un carácter ; y un comentario. Este comentario será ignorado. Los espacios y tabuladores al comienzo de cada línea son ignorados.

Además de los ficheros script propiamente, también existen ficheros de definición de macros que contienen comandos MACRO. Estos ficheros tienen extensión **.inc** y tienen que estar grabados en el disco duro de SUTIL dentro del directorio **SysScr** dentro del directorio de SUTIL (normalmente **C:\Sutil**).

Los scripts deben ser compilados para generar ficheros con el código binario que SUTIL utiliza, para generar estos ficheros compilados debemos ejecutar el comando Compilador.exe que se encuentra en el directorio de SUTIL.

Al ejecutar este comando primeramente compila todos los ficheros de definición de macros y posteriormente compila todos los scripts que encuentre tanto en el directorio **Scripts** como los que encuentre en el directorio **SysScr**. Los ficheros compilados tienen de extensión **.bin** y se guardan en el directorio **Bin**.

En el proceso de compilación por defecto todas las letras se cambian a mayúsculas, si queremos mantener algún texto tal cual, sin cambiarlo, debemos incluirlo dentro de comillas dobles, el compilador todo el texto que encuentra entre comillas dobles no lo modifica, las comillas dobles no aparecen en el fichero compilado.

Al compilar los scripts, si hay algún error, veremos un aviso del tipo de error y de la línea donde se encuentra el error, para facilitar la localización de esta línea donde esta el error, en el directorio donde se encuentre dicho script (normalmente el directorio **scripts**, aunque también puede ser el directorio **sysscr**) podremos encontrar un fichero de nombre como el scripts al que además de la extensión **.ivr** se le ha añadido la extensión **.obj** y donde las líneas están numeradas, de forma que podemos localizar rápidamente y sin duda la línea donde se encuentra el error.

Una vez solucionado el error al volver a compilar, si no hay errores, el fichero **.obj** desaparece.

Si el error esta en una línea de una macro contenida en un fichero **.inc**, el error de todos modos lo da en los ficheros **.ivr** que utilicen la macro con el error, de forma que la línea que nos indica que es la que contiene el error, veremos que viene de haber sustituido la llamada a la macro por su contenido.

### **2.1.2.- Llamadas a los ficheros script.**

En la pantalla de "líneas exteriores" se asocia a cada número de tlf. de llegada o línea RTC de llegada una extensión de operador humano y/o un nombre de script IVR. Se puede configurar en cada caso para que:

- 1.- siempre se traspase al operador humano.
- 2.- siempre se traspase al fichero IVR.

También se pueden asociar ficheros script a números de extensión. En este caso, el script se ejecutará al llamar a la extensión ya sea externamente o internamente. En el inicio de SUTIL también se ejecutan el script USERINIT.IVR caso de que exista dentro del directorio de scripts.

### **2.1.3.- Recarga de ficheros script.**

Al ejecutar Sutil, se almacenan en memoria la versión compilada de todos los scripts que encuentre en el directorio Bin, si posteriormente modificamos algún script, necesitaremos recargar en memoria la nueva versión, para ello el motor de Sutil dispone de un punto de "Recarga de ficheros scripts".

Este punto va cargando en memoria, cada uno de los scripts compilados que encuentra en el directorio Bin.

Si no existe un script con ese nombre cargado en memoria, lo carga y listo.

Si ya existe un script con ese nombre cargado en memoria:

- Si en ese momento no hay ninguna instancia ejecutándose de ese script, borra la versión antigua y deja cargada la nueva.
- Si en ese momento hay alguna instancia ejecutándose de ese script, guarda la versión actual hasta que no quede ejecutándose ninguna instancia de él.

Es importante aclarar que cuando se ejecuta una instancia de un scripts, no se puede modificar la versión del script que se este ejecutando. La nueva versión de los scripts se utilizan para las nuevas instancias que se lancen de los scripts.

### **2.1.4.- Códigos de resultado estándar.**

- 0.- Ok.
- 1.- Timeout.
- 2.- Comunicando.
- 3.- Error de sintaxis.
- 4.- Item inexistente.
- 5.- Recurso no disponible.
- 6.- Código incorrecto.

7.- Recurso en uso.

### **2.1.5.- Comandos información estado Sutil**

#### **GETINFOSERVICE Servicio NumVar**

*Descripción:* Nos da información de cuantas llamadas simultaneas hay en ese momento a *Servicio*.

Servicio puede ser un DDI, un destino VoIP SIP o un Alias de un DDI o de varios.

*Salidas anómalas:* 4.- Variable no existente (mayor que 39).

#### **GETINFOSCRIPT Script NumVar**

*Descripción:* Nos da información de cuantas instancias simultaneas hay en ese momento de *Script*. Se incluyen los script que estén detenidos a la espera de que se retorne a ellos desde un script llamado por un CALL.

*Salidas anómalas:* 4.- Variable no existente (mayor que 39).

## **2.2.- Directivas de pre-compilación.**

Las directivas MACRO sólo pueden estar en ficheros include .INC

### **:etiqueta**

*Descripción:* Especifica una etiqueta a la que se puede saltar con goto.

### **INCLUDE ficheroIVR**

*Descripción:* Inserta el fichero en la posición indicada. El fichero debe tener la extensión ".IVR", en el comando se puede indicar el nombre del fichero con la extensión o sin ella. No se permite el anidamiento de comandos INCLUDEs.

### **MACRO Comando Arg1 Arg2 ....**

**...Instrucciones...**

### **ENDM**

*Descripción:* Realiza una sustitución sintáctica cuando aparezca *Comando* en un fichero script por el contenido del comando MACRO. Los argumentos que sigan a *Comando* en el fichero script se sustituyen en el cuerpo de la macro. Estos argumentos deben figurar precedidos por \$ dentro del cuerpo de la macro.

Aquellos argumentos que sean opcionales deben ser los últimos, si se colocan menos argumentos detrás de comando que los que se incluyen en su definición., el compilador los sustituirá por una cadena vacía en el cuerpo de la macro.

## 2.3.- Comandos de control de ejecución.

La ejecución de los comandos presentes en un fichero script es secuencial y sólo varían su orden de ejecución por los comandos presentes en este apartado y por los comandos ONHOOK, ONSOCKET y RESERVEEXTENSION.

Hay dos maneras de iniciar desde un script un nuevo fichero script, los comandos CALL e INVOKE. El comando CALL permite llamar a otro proceso script como hijo del primero. El script desde donde se efectúa la llamada permanece parado hasta que el script llamado retorna. El comando INVOKE sin embargo llama al nuevo script e inicia su ejecución de forma independiente y paralela al primero. La única relación con el script llamador es que éste le ha podido pasar unos parámetros de ejecución.

Cuando durante la ejecución de un script se llega al final, si el script es un script hijo se produce automáticamente un RETURN 0 al script padre, en caso contrario se produce automáticamente un EXIT.

### EXIT

*Descripción:* Finaliza la ejecución del fichero script.

Cierra todos los canales que queden abiertos y no hayan sido asociados entre sí.

Si el script tenía bloqueada alguna variable global la desbloquea.

Un EXIT en un script hijo provoca también un EXIT en el script Padre,... .

### PAUSE [Tiempo]

*Descripción:* Espera el número de segundos indicado antes de continuar.

*Tiempo* debe estar comprendido entre 0 y 2147483647 .

Si no indicamos *Tiempo*, se hace una pausa hasta que ocurra alguna interrupción (ONHOOK, RESERVEEXTENSION u ONSOCKET)

(Esta última opción esta disponible desde v.2.13.1.0)

### GOTO *etiqueta*

*Descripción:* Desvía la ejecución a la etiqueta indicada.

### GOSUB *etiqueta*

*Descripción:* Desvía la ejecución a la etiqueta indicada, dejando la posibilidad de volver a la siguiente instrucción usando el comando RETSUB.

El tamaño máximo de la cola de direcciones de retorno es de 64.

Este comando no puede ser ejecutado dentro de un bloque SELECTCASE.

*Salidas anómalas:* 7.- Si se supera el tamaño máximo de la pila de direcciones

## RETSUB

*Descripción:* Devuelve la ejecución a la siguiente instrucción del último comando GOSUB.

### **IF cadena1 comp cadena2**

*...Instrucciones...*

**<ELSE>**

*...Instrucciones...*

### **ENDIF**

*Descripción:* Realiza una comparación. Si es verdadera, ejecuta los comandos entre IF y ELSE y si resulta falsa ejecuta los comandos entre ELSE (si es que existe) y ENDIF. Se permiten anidamientos de comandos IF.

*comp* puede ser ==, !=, >, <, >=, <=

Si el comparador es == ó != se hace una comparación de cadenas. Si el comparador es >, <, >= ó <= se interpretan como números enteros los argumentos a comparar ( para ello se van tomando caracteres desde la izquierda de cada argumento, hasta que acaba o se encuentra un carácter no numérico) y se comparan como números.

### **SELECTCASE cadena1**

**CASE comp cadena2**

*...Instrucciones...*

**CASE comp cadena3**

*...Instrucciones...*

**CASEELSE**

*...Instrucciones...*

### **ENDSELECTCASE**

*Descripción:* Realiza una comparación de un valor con distintos casos. Si la comparación en alguno de los casos es verdadera, ejecuta las instrucciones entre un CASE y el siguiente CASE o CASEELSE, y si todos los CASE resultan falsos ejecuta las instrucciones entre CASEELSE (si es que existe) y ENDSELECTCASE.



No se permite anidamientos de comandos SELECTCASE, aunque si se permite introducir comandos IF dentro de los CASE.

*comp* puede ser ==, !=, >, <, >=. <=

Si el comparador es == ó != se hace una comparación de cadenas. Si el comparador es >, <, >= ó <= se interpretan como números enteros los argumentos a comparar ( para ello se van tomando caracteres desde la izquierda de cada argumento, hasta que acaba o se encuentra un carácter no numérico) y se comparan como números.

### **IFERR *Nerr1a* [*Nerr1b*] ...**

**...Instrucciones...**

### **IFERR *Nerr2a* [*Nerr2b*] ...**

**...Instrucciones...**

...

### **ENDIFERR**

*Descripción:* Ejecuta unas instrucciones sólo en el caso de que la variable %err tome unos valores determinados. Se pueden colocar varios IFERR seguidos con un único ENDIFERR al final de todos ellos. No se permite el anidamiento de comandos IFERR.

### **CALL *FicheroIVR* *Arg1* *Arg2* ....**

*Descripción:* Ejecuta un fichero IVR ajeno. El fichero debe tener la extensión ".IVR", en el comando se puede indicar el nombre del fichero con la extensión o sin ella. El actual fichero se queda parado hasta que el llamado retorne. Los argumentos que se pasan son copiados en las variables locales %v0, %v1, ... del fichero llamado. Dichas variables locales del fichero llamado son propias de este y cualquier modificación en ellas no varía nada en el fichero original.

Los canales, sin embargo, sí son comunes a ambos ficheros, por lo que el script hijo puede utilizar los canales abiertos en el script padre y cualquier modificación sobre estos actuará de forma definitiva para los dos, y de igual forma el el scrip padre puede utilizar los canales abiertos por un script hijo.

De igual forma el valor que se haya podido definir para nuestro CallerID también afectara a las llamadas externas que se realicen desde el fichero llamado y cualquier modificación que se haga del CallerID en el fichero llamado, afectara, al volver, a las llamadas que se hagan desde el fichero superior.

Las variables del sistema %nums, %numd y %op mantiene en un script hijo el contenido que tenían en el scrip padre.

Los comandos OPENSOCKET y OPENFILE hechos en un script se mantienen en un script hijo,... o en un script padre,....

Una variable global bloqueada por un script puede ser desbloqueada por un script hijo,... o por un script padre,...

Un comando EXIT ejecutado en un script hijo, hace que se ejecute también un comando EXIT en el script padre.

Asimismo un cuelgue estando en el script hijo, hace que se ejecute lo que este configurado por un comando ONHOOK en dicho script (o los valores por defecto) y posteriormente si se ejecuta un RETURN (NO un EXIT), al volver al script padre ejecuta lo que este configurado por un comando ONHOOK en dicho script (o los valores por defecto).

El retorno de una llamada CALL puede leerse en la variable %ret. Pueden anidarse llamadas CALL entre distintos ficheros.

*Salidas anómalas:* 4.- Fichero inexistente.

## **JUMP FicheroIVR Arg1 Arg2 ....**

*Descripción:* Ejecuta un fichero IVR ajeno. El fichero debe tener la extensión ".IVR", en el comando se puede indicar el nombre del fichero con la extensión o sin ella.

El fichero actual, (script padre), se cierra anulando las instrucciones, ONHOOK, RESERVEEXTENSION y ONSOCKET que estuvieran abiertas en el momento de ejecutar el comando JUMP.

Los argumentos que se pasan son copiados en las variables locales %v0, %v1, ... del fichero llamado. Dichas variables locales del fichero llamado son propias de este y no tiene en cuenta las variables que no le hayamos pasado del fichero original.

Los canales, sin embargo, sí se transfieren al fichero llamado, por lo que el script hijo puede utilizar los canales abiertos en el script padre.

De igual forma el valor que se haya podido definir para nuestro CallerID también afectara a las llamadas externas que se realicen desde el fichero llamado.

Los comandos OPENSOCKET y OPENFILE hechos en un script se mantienen en un script hijo,...

Una variable global bloqueada por un script puede ser desbloqueada por un script hijo.

Pueden anidarse llamadas JUMP entre distintos ficheros.

*Salidas anómalas:* 4.- Fichero inexistente.

## **RETURN Valor**

*Descripción:* Retorna la ejecución a la siguiente instrucción al comando CALL previamente ejecutado para llamar al script en curso. Hace que la variable %ret tome el valor especificado por *Valor*.

Ejecutar un comando RETURN en un script que no ha sido ejecutado por un comando CALL, equivale a ejecutar un comando EXIT.

## INVOKE *Fichero*IVR *Arg1 Arg2 ....*

*Descripción:* Ejecuta un fichero IVR ajeno. El fichero debe tener la extensión ".IVR", en el comando se puede indicar el nombre del fichero con la extensión o sin ella. El actual fichero continua su ejecución normalmente. Los argumentos que se pasan son copiados en las variables locales %v0, %v1, ... del fichero llamado. Dichas variables locales del fichero llamado son propias de este y cualquier modificación en ellas no varía nada en el fichero original. Los canales y el valor que se le haya podido definir a nuestro CallerID, también son independientes, por lo que en el momento en el que comienza la ejecución del fichero invocado son totalmente independientes uno del otro.

Si uno de los parámetros que se pasa es una variable, todo el contenido de la variable pasa a la variable correspondiente en el fichero script invocado, incluso si contiene espacios en blanco.

*Salidas anómalas:* 4.- Fichero inexistente.

Ejemplos:

En el siguiente ejemplo se invoca al script Phoning.ivr con 3 parámetros.

```
INVOKE Phoning 951088500 1 31
```

En el script invocado (Phoning.ivr) la variable %w00 contendrá **951088500**, la variable %w01 contendrá **1** y la variable %w02 contendrá **31** .

Sin embargo, suponiendo que la variable %w01 contiene **Phoning** , la variable %w02 contiene **951088500** y la variable %w03 contiene **1 31** , con el siguiente comando se invoca al script Phoning.ivr con 2 parámetros.:

```
INVOKE %w01 %w02 %w03
```

En el script invocado (Phoning.ivr) la variable %w00 contendrá **951088500** y la variable %w01 contendrá **1 31** .

## SETPRIORITYSCRIPT *Priority*

Disponible desde v.2.11.2.0

Ejecuta el número de instrucciones atómicas indicadas por priority en un único ciclo de sutil, si antes de ejecutar este número de instrucciones atómicas, se llega a una instrucción no atómica, se ejecuta dicha instrucción no atómica y ya no se ejecutan más instrucciones hasta el siguiente ciclo de Sutil, en él que se ejecutan como máximo el número máximo de instrucciones atómicas por ciclo por defecto (5).

Priority puede tomar un valor entre 1 y 1000 (ambos inclusive), el valor por defecto es 5.

*Salidas anómalas:* 4.- Valor de Priority no válido.

5.- Este error se lanza si hacemos otro SETPRIORITYSCRIPT en el mismo ciclo de ejecución de sutil, el anterior no ha terminado y el número de ejecuciones que indica el argumento priority es MAYOR que el que quedan por ejecutar de la anterior ejecución de SETPRIORITYSCRIPT. Lo que no provoca una salida anómala es bajar la anterior prioridad.

## 2.4.- Gestión de variables.

Las variables siempre se tratan como cadenas de caracteres internamente. Hay cuatro tipos de variables: de usuario locales numéricas, de usuario locales definidas, de usuario globales y del sistema.

Las variables de usuario locales numéricas son: %v0, %v1, ... %v9, %w00, %w02, ... %w39. Cuando usamos %v solo podemos poner una cifra detrás ( por eso solo podemos poner del 0 al 9 ) y si utilizamos %w tenemos que poner 2 cifras, pero la variable %v0 es la misma que la variable %w00 y así sucesivamente hasta la variable %v9, que es la misma que la variable %w09.

Las variables de usuario locales definidas, son variables que el usuario a declarado previamente con un nombre cualquiera. Como máximo podremos definir 60 de estas variables en un mismo script.

Las variables de usuario globales son %g0, %g1, ... %g9.

Y las variables del sistema son:

%err Resultado del último comando.

%suberr Sub-resultado del último comando. Si un resultado principal de un comando, (%err), puede ser originado por varios motivos, en esta variable distinguiremos entre las distintas causas que han provocado el resultado principal, (%err).

%numd Destino donde ha llegado la llamada, esta variable solo tiene sentido en script ejecutados para atender una llamada entrante.

En una Línea RDSI Destino es: Número de teléfono para el que viene la llamada.

En una Línea Analógica Destino es: Número de teléfono asignado a la línea por la que entra la llamada.

En un Recurso VoIP mediante protocolo SIP Destino es: Contenido a la izquierda de la @ del usuario SIP indicado en el campo *To* de los paquetes SIP.

%nums Origen desde el que nos ha llegado la llamada, esta variable solo tiene sentido en script ejecutados para atender una llamada entrante.

En una Línea RDSI Origen es: Número de teléfono desde el que nos ha llegado la llamada, en las llamadas ocultas este valor esta vacío.

En una Línea Analógicas Origen es: Número de teléfono desde el que nos ha llegado la llamada, en las llamadas ocultas este valor esta vacío.

En un Recurso VoIP mediante protocolo SIP Origen es:

%phandle Handle único para cada script ejecutándose que lo identifica.

%session Handle del script principal. En el caso de que el script actual haya sido llamado con el comando CALL, esta variable contiene el handle del script padre original que originó las llamadas.

%source Si en la memoria del sistema existe el Número de teléfono origen de la llamada, contiene el Nombre asociado a dicho número y si no, contiene el Numero de telefono origen de la llamada.

<code>%op</code>	Operador por defecto, esta variable contiene el valor indicado en la columna <i>Operador</i> en la configuración de los scripts que atienden las llamadas.
<code>%Sutil</code>	Número, dentro del Dominio al que pertenezca, del Sutil donde se esta ejecutando el script, este número se define en el fichero Sutil.ini, si no se ha definido se devuelve el valor 1, porque se entiende que solo hay un Sutil.
<code>%ret</code>	Valor retornado en una llamada con CALL.
<code>%timespent</code>	Número de segundos que han pasado desde el inicio del script.
<code>%date</code>	Fecha actual en el formato: aaaa/mm/dd
<code>%time</code>	Hora actual en el formato hh:mm:ss
<code>%day</code>	Dia de la semana, 1 = lunes, ... 7 = domingo

En cualquier parte de un fichero script se puede colocar una variable que será sustituida por su valor antes de evaluar el comando. Por tanto para concatenar cadenas simplemente basta con yuxtaponerlas.

Por ejemplo, si `%v0="12"` y `%v1="hola"`, si en un comando aparece como argumento:

```
hh%v023%v1
```

el argumento valdrá:

```
Hh1223hola
```

Las variables de usuario locales definidas, una vez declaradas tienen las mismas propiedades, y podemos hacer con ellas las mismas funciones que con cualquiera de las variables de usuario locales numéricas.

Es aconsejable que a la hora de definir variables que el nombre que le asignemos nos pueda orientar sobre cual va a ser el contenido de la variable.

Para definir una variable usamos el comando "VAR NombreVariable", y para conocer el valor que contiene la variable usamos "%NombreVariable".

La diferencia entre variables locales y globales está en que las locales son propias de cada ejecución del script y su modificación en un script no afecta al valor que se pueda leer en otra ejecución de ese o de otro script, mientras que las globales si tienen un valor común para todas las ejecuciones de scripts.

Para sincronizar el acceso a las variables globales se dispone del comando LOCKGLOBAL que permite bloquear el acceso a una variable global por parte de otros procesos.

Aunque las variables se tratan internamente como cadenas, se dispone del comando EVAL que permite evaluar una expresión aritmética que incluya variables y/o números enteros.

## **VAR NomVar**

*Descripción:* Declara la variable local definida con el nombre NomVar. Como máximo podremos definir 60 de estas variables en un mismo script.

En el siguiente ejemplo definimos una variable para usarla como un contador.

```
VAR Contador
SETLOCAL Contador 0
IF %Contador <= 5
    INLOCAL Contador
ENDIF
```

### **SETLOCAL NumVar [Cadena]**

*Descripción:* Asigna a la variable local indicada por NumVar el valor de *Cadena*. Esta *Cadena* puede contener espacios en blanco, en cualquier caso se le asigna a la variable todo el contenido después de NumVar hasta el final de la línea.

Si no se incluye *Cadena*, se le asigna a la variable una cadena vacía.

*Salidas anómalas:* 4.- Variable no existente (mayor que 39).

### **DECLOCAL NumVar**

*Descripción:* Decrementa el valor contenido en la variable local indicada por NumVar. Si la variable contiene un valor no numérico, se interpreta como 0, pasando a tomar el valor -1.

*Salidas anómalas:* 4.- Variable no existente (mayor que 39).

### **INCLOCAL NumVar**

*Descripción:* Incrementa el valor contenido en la variable local indicada por NumVar. Si la variable contiene un valor no numérico, se interpreta como 0, pasando a tomar el valor 1.

*Salidas anómalas:* 4.- Variable no existente (mayor que 39).

### **EVAL NumVarLoc ExpAlgebraica**

*Descripción:* Evalúa una expresión algebraica que puede utilizar +, -, \* ó / de números enteros y/o variables, se pueden utilizar paréntesis, se pueden insertar espacios en blanco dentro de la expresión algebraica para hacer mas clara su interpretación. Tanto el resultado final como los intermedios se evalúan como enteros. La cantidad resultante se asigna a la variable local indicada.

*Salidas anómalas:* 4.- Variable no existente (mayor que 39).

Ejemplos:

En el siguiente ejemplo se le asigna a la variable 8 el resultado de restarle a 3 el contenido de la variable 5.

```
EVAL 8 3 - %w05
```

En el siguiente ejemplo se le asigna a la variable 8 el resultado de multiplicar el contenido de la variable 5 por el contenido de la variable 6.

```
EVAL 8 %w05 * %w06
```

## **SETGLOBAL NumVar [Cadena]**

*Descripción:* Asigna a la variable global indicada por NumVar el valor de Cadena. Esta Cadena puede contener espacios en blanco, en cualquier caso se le asigna a la variable todo el contenido después de NumVar hasta el final de la línea.

Si no se incluye Cadena, se le asigna a la variable una cadena vacía.

Si la variable está previamente bloqueada, se desbloquea.

*Salidas anómalas:* 4.- Variable no existente (mayor que 9).  
7.- Variable bloqueada por otro hilo de ejecución.

## **LOCKGLOBAL NVar Timeout**

*Descripción:* Bloquea la variable global indicada por Nvar.

La variable se desbloqueará por uno de los siguientes motivos:

- En el momento que un script con el mismo %session (puede ser el mismo script o un script padre,... o hijo,...) escriba en ella con los comandos SETGLOBAL, DECGLOBAL o INCGLOBAL.
- Al terminar la ejecución del último script padre, ... del script que la bloqueo, o del mismo script en caso de que no tenga ningún script padre.

Mientras está bloqueada, ningún otro hilo de ejecución podrá escribir en ella, ni bloquearla.

Si al intentar bloquear la variable ya esta bloqueada por otro hilo de ejecución, se reintenta el tiempo indicado por Timeout antes de dar un error 5.

*Salidas anómalas:* 4.- Variable no existente (mayor que 9).  
5.- Variable bloqueada por otro hilo de ejecución.

## **DECGLOBAL NumVar**

*Descripción:* Decrementa el valor contenido en la variable global indicada por NumVar. Si la variable contiene un valor no numérico, se interpreta como 0, pasando a tomar el valor -1. Si la variable está previamente bloqueada, se desbloquea.

*Salidas anómalas:* 4.- Variable no existente (mayor que 9).  
7.- Variable bloqueada por otro hilo de ejecución.



**INCGLOBAL NumVar**

*Descripción:* Incrementa el valor contenido en la variable global indicada por NumVar. Si la variable contiene un valor no numérico, se interpreta como 0, pasando a tomar el valor 1. Si la variable está previamente bloqueada, se desbloquea.

*Salidas anómalas:* 4.- Variable no existente (mayor que 9).  
7.- Variable bloqueada por otro hilo de ejecución.

**STRSUB NumVarDest Origen Longitud Cadena**

*Descripción:* Copia en la variable destino un sub-segmento de la longitud indicada de la cadena suministrada empezando en el carácter indicado por *Origen*.

Cuando *Origen* vale cero se empieza en el primer carácter y así sucesivamente.

Si *Longitud* es superior a los caracteres existentes desde *Origen* hasta el final de la *Cadena*, devuelve estos caracteres existentes desde *Origen* hasta el final de la *Cadena*, sin añadirle espacios en blanco ni ningún otro carácter de relleno.

*Salidas anómalas:* 4.- Variable no existente (mayor que 39).

**STRTOKEN NumVarDest NumToken Separador Cadena**

*Descripción:* Descompone la cadena suministrada en subcadenas utilizando el separador indicado y almacena en la variable indicada el número de subcadena indicada por NumToken.

Si se indica un *NumToken* superior al número de Tokens contenidos en *Cadena*, se devuelve un error 5 y la variable indicada por *NumVarDest* vacía.

Los separadores válidos son:

Cualquier letra en mayúscula o minúscula.

Cualquier número.

! # \$ % \* , - . / : < = > @ ^ |

*Salidas anómalas:* 4.- Variable no existente (mayor que 39).  
5.- Token inexistente.

**STRTOKENMULT NumVarInicial MaxTokens NumVarLeidos Sep Cadena**

*Descripción:* Descompone la cadena suministrada en subcadenas utilizando el separador indicado y almacenándolas en las variables indicadas desde NumVarInicial hasta (NumVarInicial + MaxTokens).

*NumVarInicial* es el número de variable a partir de la cual se guardan los tokens.

*MaxTokens* es el número máximo de tokens a leer, si hay más tokens no se leen.

NumVarLeidos es el número de variable en la que se guarda el número de tokens que realmente se han leído.

Los separadores válidos son:

Cualquier letra en mayúscula o minúscula.

Cualquier número.

! # \$ % \* , - . / : < = > @ ^ |

*Salidas anómalas:* 4.- Numero de Variable inexistente.

Ejemplo:

El siguiente ejemplo desglosa los 3 primeros tokens de una cadena, el separador utilizado es "," (coma).

Suponemos que la variable %w01 contiene:

1245,951900900,4,Orange

STRTOKENMULT 5 3 2 , %w01

A continuación indicamos el contenido de las distintas variables después de ejecutar este comando:

%w02	3
%w05	1245
%w06	951900900
%w07	4

## **SETERROR *Nerr***

*Descripción:* Fija la variable %err con el valor indicado.

*Salidas anómalas:* %err toma el valor indicado.

## **RANDOM *Desde Hasta Numvar***

*Descripción:* Asigna a la variable local especificada por *Numvar* un número aleatorio entre *Desde* y *Hasta* (ambos inclusive).

## **CMPTIME *Time Numvar***

*Descripción:* Compara *Time* con la hora actual, y en *Numvar* nos devuelve:

-1 si *Time* es menor que la hora actual.  
0 si *Time* es igual a la hora actual.  
1 si *Time* es mayor que la hora actual.

*Time* debe estar en el formato hh:mm:ss .

**CMP2TIME *Time1 Time2 Numvar***

*Descripción:* Compara *Time1* con *Time2*, y en *Numvar* nos devuelve:

- 1 si *Time1* es menor que *Time2*.
- 0 si *Time1* es igual que *Time2*.
- 1 si *Time1* es mayor que *Time2*.

*Time1* y *Time2* deben estar en el formato hh:mm:ss .

**INTTIME *TimeFrom TimeTo Numvar***

*Descripción:* Comprueba si la hora actual esta dentro del intervalo indicado, y en *Numvar* nos devuelve:

- 1 si *TimeFrom* <= Hora actual <= *TimeTo*
- 0 en otro caso.

*TimeFrom* y *TimeTo* deben estar en el formato hh:mm:ss .

**INTDATE *DateFrom DateTo NumVar***

*Descripción:* Comprueba si la fecha actual esta comprendida dentro del intervalo indicado, y en *Numvar* nos devuelve:

- 1 si *DateFrom* <= FechaActual <= *DateTo*
- 0 en otro caso.

*DateFrom* y *DateTo* deben estar en el formato aaaa/mm/dd .

**CONVDATE *Date NumVar***

*Descripción:* Convierte una fecha en formato aaaa/mm/dd en un formato numérico aaaammdd

Argumentos:

Date - Fecha que se va a transformar

NumVar - Número de la variable donde se retorna el resultado:

**CONVTIME *Time NumVar***

*Descripción:* Convierte una hora en formato hh:mm:ss en un formato numérico hhmmss

Argumentos:

Time - Fecha que se va a transformar

NumVar - Número de la variable donde se retorna el resultado:

## **MOD *Dividendo Divisor NumVar***

Descripción: Realiza la función mod

Argumentos:

Dividendo - Número al que le vamos a hacer la división

Divisor - Divisor por el cual vamos a dividir

NumVar - Número de la variable donde se retorna el resultado:

## 2.5.- Gestión y conexión de canales.

Todos los comandos de reproducción (PLAY) y grabación (RECORD e INPUT) se pueden reproducir a través de distintos canales. El canal por el que serán utilizados más habitualmente es por el de la llamada entrante, que siempre será el canal 0.

Cuando entra una llamada al sistema y se pone en marcha un script, esta llamada se asocia con el canal 0. Sin embargo, la llamada no se responde o rechaza hasta que se ejecuta el comando INCOMINGCALL y sólo a partir de ese momento podemos reproducir o grabar en dicho canal.

Durante la ejecución del script, se pueden abrir otros canales hacia extensiones o líneas externas, donde también se pueden ejecutar estos comandos. Estos canales se abren mediante los comandos CALL2EXTENSION y CALL2EXTERIOR.

Una vez abiertos pueden ser colgados con el comando HOOK o enlazados entre sí con el comando CONNECT. Al finalizar el script se colgarán automáticamente aquellos canales que no estén entrelazados.

Los canales válidos son del 0 a 19, ambos inclusive, aunque el canal 0 solo debe utilizarse para el canal entrante al que se asocia automáticamente, no debemos utilizarlo para las llamadas que haga el script.

Para poder conectar varios canales en modo multiconferencia hay que hacerlos conectándolos todos a un salón (room) a través del comando CONECT2ROOM.

### **INCOMINGCALL Acción [Causa]**

*Descripción:* Cuando entra una llamada al sistema y se pone en marcha un script, esta llamada se asocia con el canal 0. Sin embargo, la llamada no se acepta, rechaza o se le indica que avanza la llamada, hasta que se ejecuta este comando.

*Acción* puede valer:

- 0.- Se acepta la llamada.
- 1.- Se rechaza la llamada
- 2.- Se envía un Alert indicando que la llamada se esta cursando.

Si la llamada se recibe por una línea analógica, solo es válida la primera acción.

En el caso de de que rechazemos la llamada podemos enviar la causa por la que se rechaza, según la tabla de causas por las que no tiene éxito una llamada según el protocolo RDSI.

La ejecución de este comando con la opción 1 no provoca el salto a la etiqueta que pudiera estar configurado con un comando ONHOOK.

Solo se puede utilizar la acción de enviar un Alert en el caso de que tengamos configurado *AutoAlert = 0* en el fichero Sutil.ini.

Si tenemos configurado *AutoAlert = 0* en el fichero *Sutil.ini*, podemos ejecutar primero este comando con *Accion = 2* y posteriormente volverlo a ejecutar con *Accion = 0* o directamente ejecutarlo con *Accion = 0* sin haberlo ejecutado previamente con *Accion = 2*.

*Salidas anómalas:*                      6.- Código incorrecto.

## **CALL2EXTENSION Canal NExt Timeout [MensajeLCD]**

*Descripción:* Realiza una llamada a una extensión o grupo y la asocia al canal especificado por *Canal*. Si la extensión no responde o comunica, el sistema no pasa automáticamente a la opción de dejar un mensaje. Si se quiere que esto ocurra deberá ser implementado con los correspondientes comandos dentro del fichero *SCRIPT*.

*Canal* es el canal que se abrirá, al que se asociará la extensión, debe tener un valor entre 1 y 9.

*NExt* es el número de extensión o grupo al que se intenta llamar.

*Timeout* indica el máximo tiempo en segundos a esperar, este valor debe estar comprendido normalmente entre 20 y 40. Aproximadamente cada 3 seg. equivalen a un Ring.

En el caso de que la extensión este autenticada en un teléfono externo, un valor por debajo de 20 puede ser demasiado pequeño de forma que puede que la llamada no llegue a sonar en el destino, sobre todo en el caso de llamadas a móviles o al extranjero. Un valor superior a 40 seg. puede no tener sentido, ya que normalmente las Operadoras tienen un tiempo máximo, para que tenga éxito la llamada, de alrededor de 40 sg., de forma que la Operadora daría la llamada por fallida antes del *Timeout* indicado por nosotros.

*MensajeLCD* es una cadena que se mostrará en el display lcd de la extensión, el tamaño máximo es 32 caracteres (incluyendo el carácter 0 de fin de cadena), si es mayor se trunca.

Cuando la extensión este autenticada en un teléfono externo, se utiliza la *Política de LCR 0* (cero) para seleccionar la línea a utilizar para realizar la llamada saliente, si no hay definida una política de LCR 0 para llamar al número de teléfono en cuestión, Sutil realiza la llamada utilizando de forma cíclica (Round Robin) todas las líneas existentes (tanto analógicas como digitales).

Si se ha indicado un grupo en vez de una extensión, Sutil decide a que extensión del grupo debe llamar según como se haya configurado la distribución de las llamadas al grupo. Si la extensión a la que corresponde llamar es una extensión externa y al llamarla comunica, Sutil intenta llamar a otra extensión, mientras quede alguna disponible a la que no se haya llamado.

Si el grupo tiene el parámetro *RetryCall* configurado a 1 ( en el fichero *Sutil.ini* ), si la extensión a la que se llama no contesta, Sutil llama a otra, mientras quede alguna disponible a la que no se haya llamado.

*Salidas anómalas:*                      1.- Timeout.

2.- Si se esta llamando a una extensión: la extensión esta comunicando o no esta activa.

Si se esta llamando a un grupo: todas las ext. del grupo están comunicando o no están activas.

4.- Si se esta llamando a una extensión: la extensión no existe o no esta autenticada.

Si se esta llamando a un grupo: el grupo no existe o no hay en ese momento ninguna extensión del grupo autenticada (incluyendo el caso de que el grupo no tenga definida ninguna extensión).

5.- No hay líneas disponibles para llamar (solo posible en el caso de extensiones externas).

7.- Canal en uso.

### **CALL2EXTENSIONBG Canal NvarEstado Next CanalSecEsc SecEsc Timeout [MensajeLCD]**

*Descripción:* Realiza una llamada a una extensión o grupo en BackGround y la asocia al canal especificado, una vez se ejecuta esta instrucción la ejecución del script continua, mientras se esta realizando la llamada el script debe ser el encargado de vigilar el avance de la llamada, consultando la variable NvarEstado y tomar las medias oportunas (poner tono de ocupado o llamando, etc...).

La llamada deja de realizarse si se comprueba que la extensión llamada comunica, contesta, si se recibe la secuencia de escape por el canal indicado o cuando vence el timeout.

*Canal* es el canal que se abrirá, al que se asociará la llamada, debe tener un valor entre 1 y 9.

*NvarEstado* es el número de variable donde nos va devolviendo el estado en el que se encuentra la llamada:

0.- Progresando.

1.- Este valor de estado no se devuelve nunca.

2.- Si se esta llamando a una extensión: la extensión comunica o no esta activa.

Si se esta llamando a un grupo: todas las ext. del grupo están comunicando o no están activas.

3.- Conectado.

4.- Se ha pulsado la secuencia de Escape.

5.- Timeout

*NExt* es el número de extensión o grupo al que se intenta llamar.

*CanalSecEsc* indica por que canal se espera recibir la secuencia de escape.

*SecEsc* indica cual es la secuencia de escape, esta secuencia de escape puede ser uno o más digitos, si se pulsan estos dígitos esta instrucción se detiene y en la variable *NvarEstado* se indica dicha circunstancia. Si no queremos poner una secuencia de escape debemos poner X.

*Timeout* indica el máximo tiempo en segundos a esperar, este valor debe estar comprendido normalmente entre 20 y 40. Aproximadamente cada 3 seg. equivalen a un Ring.

En el caso de que la extensión este autenticada en un teléfono externo, un valor por debajo de 20 puede ser demasiado pequeño de forma que puede que la llamada no llegue a sonar en el destino, sobre todo en el caso de llamadas a móviles o al extranjero. Un valor superior a 40 seg. puede no tener sentido, ya que normalmente las Operadoras tienen un tiempo máximo, para que tenga éxito la llamada, de alrededor de 40 sg., de forma que la Operadora daría la llamada por fallida antes del *Timeout* indicado por nosotros.

*MensajeLCD* es una cadena que se mostrará en el display lcd de la extensión, el tamaño máximo es 32 caracteres (incluyendo el carácter 0 de fin de cadena), si es mayor se trunca.

Cuando la extensión este autenticada en un teléfono externo, se utiliza la *Política de LCR 0* (cero) para seleccionar la línea a utilizar para realizar la llamada saliente, en el caso de no haber definida una *Política de LCR 0* que incluya al teléfono a llamar, se van utilizando sucesivamente todas las líneas disponibles (tanto analógicas como digitales).

- Salidas anómalas:*
- 4.- Si se esta llamando a una extensión: la extensión no existe o no esta autenticada.  
Si se esta llamando a un grupo: el grupo no existe o no hay en ese momento ninguna extensión del grupo autenticada (incluyendo el caso de que el grupo no tenga definida ninguna extensión).
  - 5.- No hay líneas disponibles para llamar (solo posible en el caso de extensiones externas).
  - 7.- Canal en uso.

#### Ejemplo:

El siguiente ejemplo llama a la extensión 200 en background, y va comprobando el estado de la llamada, mientras avanza la llamada le pone tono de llamando durante 3 seg. y después le reproduce un mensaje.

```
SETLOCAL 11 %timespent
SETLOCAL 13 0
CALL2EXTENSIONBG 1 10 200 0 # 20
PLAYTONE 0 0
:CICLO
```

; Si comunica se le pone un tono de comunicando al llamante 5 seg. y acaba el script.

```
IF %w10 == 2
    PLAYTONE 0 2
    PAUSE 5
    EXIT
ENDIF
```

; Si descuelga la extensión llamada, conecta los 2 canales y termina el script.

```
IF %w10 == 3
```



```

CONNECT 0 1
EXIT
ENDIF

```

; Si se pulsa la secuencia de escape (#) se va a la etiqueta INICIO.

```

IF %w10 == 4
GOTO INICIO
ENDIF

```

; Si no descuelga la extensión llamada en el tiempo previsto (20.seg), termina el script.

```

IF %w10 == 5
EXIT
ENDIF

```

```

EVAL 12 %timespent - %w11
IF %w12 > 3
GOTO MENSAJE
ENDIF
GOTO CICLO

```

```
:MENSAJE
```

```

IF %w13 == 0
PLAYFILE 0 Mensaje 3
SETLOCAL 13 1
ENDIF
GOTO CICLO

```

## **CALL2EXTERIOR Canal NTeléfono Timeout [NLinea1] [NLinea2] ...**

*Descripción:* Hace una llamada a un número de teléfono externo y la asocia al canal especificado.

*Canal* es el canal que se abrirá, al que se asociará la llamada, debe tener un valor entre 1 y 9.

*Nteléfono* es el número de teléfono al que se intenta llamar.

*Timeout* indica el máximo tiempo en segundos a esperar, este valor debe estar comprendido normalmente entre 20 y 40, un valor por debajo de 20 puede ser demasiado pequeño de forma que puede que la llamada no llegue a sonar en el destino, sobre todo en el caso de llamadas a móviles o al extranjero. Un valor superior a 40 seg. puede no tener sentido, ya que normalmente las Operadoras tienen un tiempo máximo, para que tenga éxito la llamada, de alrededor de 40 sg., de forma que la Operadora daría la llamada por fallida antes del *Timeout* indicado por nosotros. Aproximadamente cada 4,5 seg. equivalen a un Ring.

*NLinea1 Nllinea2 ...* indica por qué enlace(s) se debe intentar efectuar la llamada.

Estos valores pueden ser:

- Un número que indica una *Línea* , incluyendo analógicas y digitales. Si alguna de los números de línea pertenece a un enlace RDSI se intentará hacer por cualquiera de los canales del mismo.
- **Lx**, de esta forma hacemos referencia directamente a la línea analógica número x.
- **Bx**, de esta forma hacemos referencia directamente a todos los canales del BRI número x.
- **Px**, de esta forma hacemos referencia directamente a todos los canales del PRI número x.
- **Gx**, de esta forma hacemos referencia a un grupo de líneas definido en la sección *Trunk lines groups*, en el *Manual del Administrador de Sutil* podemos consultar como se definen los grupos de líneas.
- **Sx**, de esta forma hacemos referencia al Servidor de enrutamiento x.

En el caso de que ninguna de las líneas indicadas estuviera disponible para realizar la llamada (por estar ocupadas o averiadas), no lo intenta por otras que pudieran existir.

En el caso de que no se especifiquen estos parámetros, se utiliza la *Política de LCR 0* para seleccionar la línea a utilizar para realizar la llamada, y en el caso de no haber definida una *Política de LCR 0* que incluya al teléfono a llamar, se van utilizando sucesivamente todas las líneas disponibles (tanto analógicas como digitales).

*Salidas anómalas:*

- 1.- Timeout.
- 2.- Comunicando.
- 5.- No hay líneas disponibles para llamar.
- 7.- Canal en uso.

### **CALL2EXTERIORBG Canal NvarEstado NvarCausa NTelefono CanalSecEsc SecEsc Timeout [NLinea1] [NLinea2] ...**

*Descripción:* Hace una llamada a un número de teléfono externo en Background y la asocia al canal especificado, una vez se ejecuta esta instrucción la ejecución del script continua, mientras se esta realizando la llamada el script debe ser el encargado de vigilar el avance de la llamada, consultando la variable NvarEstado y tomar las medias oportunas (poner tono de ocupado o llamando, descolgar, rechazar la llamada, etc...).

La llamada deja de realizarse cuando la variable *NvarEstado* toma uno de los valores 2, 3, 4 ó 5, o cuando se realiza un HOOK de *Canal*.

*Canal* es el canal que se abrirá, al que se asociará la llamada, debe tener un valor entre 1 y 9.

*NvarEstado* es el número de variable donde nos va devolviendo el estado en el que se encuentra la llamada:

0.- Progresando.

- BRIs).
- 1.- Alerta (solo puede aparecer en el caso de líneas digitales, PRIs o BRIs).
  - 2.- La llamada no ha tenido éxito.
  - 3.- Conectado.
  - 4.- Se ha pulsado la secuencia de Escape.
  - 5.- Timeout

*NvarCausa* es el número de variable donde nos devuelve la causa RDSI en el caso de que la llamada no haya tenido éxito.

*Nteléfono* es el número de teléfono al que se intenta llamar.

*CanalSecEsc* indica por que canal se espera recibir la secuencia de escape.

*SecEsc* indica cual es la secuencia de escape, esta secuencia de escape puede ser uno o más dígitos, si se pulsan estos dígitos esta instrucción se detiene y en la variable *NvarEstado* se indica dicha circunstancia. Si no queremos poner una secuencia de escape debemos poner X.

*Timeout* indica el máximo tiempo en segundos a esperar, este valor debe estar comprendido normalmente entre 20 y 40, un valor por debajo de 20 puede ser demasiado pequeño de forma que puede que la llamada no llegue a sonar en el destino, sobre todo en el caso de llamadas a móviles o al extranjero. Un valor superior a 40 seg. puede no tener sentido, ya que normalmente las Operadoras tienen un tiempo máximo, para que tenga éxito la llamada, de alrededor de 40 sg., de forma que la Operadora daría la llamada por fallida antes del *Timeout* indicado por nosotros. Aproximadamente cada 4,5 seg. equivalen a un Ring.

*NLinea1 Nllinea2 ...* indica por qué enlace(s) se debe intentar efectuar la llamada.

Estos valores pueden ser:

- Un número que indica una *Línea* , incluyendo analógicas y digitales. Si alguna de los números de línea pertenece a un enlace RDSI se intentará hacer por cualquiera de los canales del mismo.
- **L***x*, de esta forma hacemos referencia directamente a la línea analógica número x.
- **B***x*, de esta forma hacemos referencia directamente a todos los canales del BRI número x.
- **P***x*, de esta forma hacemos referencia directamente a todos los canales del PRI número x.
- **G***x*, de esta forma hacemos referencia a un grupo de líneas definido en la sección *Trunk lines groups*, en el *Manual del Administrador de Sutil* podemos consultar como se definen los grupos de líneas.
- **S***x*, de esta forma hacemos referencia al Servidor de enrutamiento x.

En el caso de que ninguna de las líneas indicadas estuviera disponible para realizar la llamada (por estar ocupadas o averiadas), no lo intenta por otras que pudieran existir.

En el caso de que no se especifiquen estos parámetros, se utiliza la política de LCR 0 para seleccionar la línea a utilizar para realizar la llamada, y en el caso de no haber definida una *Política de LCR 0* que incluya al teléfono a llamar, se van utilizando sucesivamente todas las líneas disponibles (tanto analógicas como digitales).

*Salidas anómalas:*

- 4.- Canal o nº de variables no válidos.
- 5.- No hay líneas disponibles para llamar.
- 7.- Canal en uso.

## **CALL2EXTERIORISDN Canal Nteléfono Timeout NvarCausa [NLinea1] [NLinea2] ...**

*Descripción:* Hace una llamada a un número de teléfono externo y la asocia al canal especificado, este comando es similar al comando *CALL2EXTERIOR*, al que se le añadido una variable para que nos devuelva la causa RDSI en el caso de no tener éxito la llamada, aunque este comando esta pensado para ser utilizado con líneas digitales, también puede ser utilizado con líneas analógicas.

*Canal* es el canal que se abrirá, al que se asociará la llamada, debe tener un valor entre 1 y 9.

*Nteléfono* es el número de teléfono al que se intenta llamar.

*Timeout* indica el máximo tiempo en segundos a esperar, este valor debe estar comprendido normalmente entre 20 y 40, un valor por debajo de 20 puede ser demasiado pequeño de forma que puede que la llamada no llegue a sonar en el destino, sobre todo en el caso de llamadas a móviles o al extranjero. Un valor superior a 40 seg. puede no tener sentido, ya que normalmente las Operadoras tienen un tiempo máximo, para que tenga éxito la llamada, de alrededor de 40 sg., de forma que la Operadora daría la llamada por fallida antes del *Timeout* indicado por nosotros. Aproximadamente cada 4,5 seg. equivalen a un Ring.

*NvarCausa* es el número de variable donde nos devuelve la causa RDSI en el caso de que la llamada no haya tenido éxito, en el caso de que la llamada no se haga por una línea digital devuelve siempre 0 en esta variable.

*NLinea1 Nllinea2 ...* indica por qué enlace(s) se debe intentar efectuar la llamada.

Estos valores pueden ser:

- Un número que indica una *Línea*, incluyendo analógicas y digitales. Si alguna de los números de línea pertenece a un enlace RDSI se intentará hacer por cualquiera de los canales del mismo.
- **Lx**, de esta forma hacemos referencia directamente a la línea analógica número x.
- **Bx**, de esta forma hacemos referencia directamente a todos los canales del BRI número x.

- **P<sub>x</sub>**, de esta forma hacemos referencia directamente a todos los canales del PRI número x.
- **G<sub>x</sub>**, de esta forma hacemos referencia a un grupo de líneas definido en la sección *Trunk lines groups*, en el *Manual del Administrador de Sutil* podemos consultar como se definen los grupos de líneas.
- **S<sub>x</sub>**, de esta forma hacemos referencia al Servidor de enrutamiento x.

En el caso de que ninguna de las líneas indicadas estuviera disponible para realizar la llamada (por estar ocupadas o averiadas), no lo intenta por otras que pudieran existir.

En el caso de que no se especifiquen estos parámetros, se utiliza la *Política de LCR 0* para seleccionar la línea a utilizar para realizar la llamada, y en el caso de no haber definida una *Política de LCR 0* que incluya al teléfono a llamar, se van utilizando sucesivamente todas las líneas disponibles (tanto analógicas como digitales).

*Salidas anómalas:*

- 1.- Timeout.
- 2.- Llamada sin éxito.
- 5.- No hay líneas disponibles para llamar.
- 7.- Canal en uso.

## **CALLSIMULTANEOUSBG Canal NvarEstado NvarCausa CanalSecEsc SecEsc Modo TimeOut ListaNumerosyCanales**

*Descripción:* Realiza de forma simultanea una llamada a cada uno de los números contenidos en *ListaNumerosyCanales*, además en esta lista también se indican, para cada número a llamar, el canal o canales a utilizar. Cuando en una de las llamadas se recibe un ALERT o descuelga el llamado ( según *Modo* ) se abortan el resto de llamadas.

Las llamadas se realizan en BackGround, es decir que una vez se ejecuta esta instrucción la ejecución del script continua, mientras se esta realizando la llamada el script debe ser el encargado de vigilar el avance de la llamada, consultando la variable *NvarEstado* y tomar las medias oportunas (poner tono de ocupado o llamando, descolgar, rechazar la llamada, etc...).

Si la llamada tiene éxito se asocia al *Canal*.

**Canal** es el que canal que se intenta abrir con esta llamada

**NvarEstado** numero de la vble en la que se guarda el estado de la llamada:

- 0.- Llamada en curso.
- 1.- Recibida alerta de llamada.
- 2.- Comunican todas las llamadas.
- 3.- Conectada.
- 4.- Recibida secuencia de escape. Fin de todas las llamadas.
- 5.- Time out. Fin de todas las llamadas.

**NvarCausa** es el número de variable donde nos devuelve la causa RDSI en el caso de que la llamada no haya tenido éxito.

En el caso de que se hayan hecho varias llamadas simultáneamente, la causa RDSI que nos devuelve es la última recibida.

**CanalSecEsc** es el que canal sobre el que se intenta recibir una posible secuencia de escape.

**SecEsc** es la secuencia de escape (X para deshabilitar).

**Modo** puede valer 0 o 1. Se utiliza para configurar cuando considerar que una llamada ya tiene éxito y hay que abortar el resto. Si vale 0 esto ocurre cuando se recibe el ALERT en una llamada y si vale 1 se aborta el resto de llamadas cuando descuelgan en el otro extremo en una de las llamadas. Si se pone un valor que no es ni 0 ni 1 se toma el valor 0.

**TimeOut** es el timeout de la llamada.

**ListaNumerosyCanales** es una lista de números de teléfono y canales por donde llamar separados por comas (,) , aunque indicamos un canal en realidad hacemos referencia a todos los canales del PRI al que pertenece dicho canal.

Los grupos de canales pueden ser o un canal o varios canales separados por el signo +. Se trata de llamar a cada uno de los números de teléfonos por su correspondiente grupo de canales. El sistema llamará dentro del grupo de canales por la primera línea que encuentre libre.

La ListaNumerosyCanales debe pasarse como un solo argumento. Por lo tanto se debe guardar el contenido en una variable y pasar dicha variable como argumento.

*Salidas anómalas:*

4.- Se está utilizando una secuencia de escape sobre un canal inexistente o no abierto.

ListaNumerosyCanales vacía.

5.- No hay líneas disponibles para llamar.

7.- Canal en uso.

Ejemplos:

Supongamos un Sutil que dispone de 2 PRIs y que *ListaNumerosyCanales* contiene:

951088500 1, 951088500 31

Se lanzará una llamada al número de teléfono 951088500 por el primer PRI y al mismo número de teléfono por el segundo PRI, una vez que una de las 2 llamadas tenga éxito se aborta la otra.

Supongamos un Sutil que dispone de 4 PRIs y que *ListaNumerosyCanales* contiene:

951088500 1, 952910400 31, 952653516 91

Se lanzara una llamada al número de teléfono 951088500 por el primer PRI, otra llamada al número de teléfono 952910400 por el segundo PRI y una tercera llamada al número de teléfono 952653516 por el cuarto PRI, una vez que una de las 3 llamadas tenga éxito se abortarán las otras dos.

Supongamos un Sutil que dispone de 4 PRIs y que *ListaNumerosyCanales* contiene:

951088500 1+31+61+91, 952910400 1+31+61+91, 952653516 1+31+61+91

Se lanzara una llamada al número de teléfono 951088500 por uno de los 4 PRIs, otra llamada al número de teléfono 952910400 por uno de los 4 PRIs y una tercera llamada al número de teléfono 952653516 por uno de los 4 PRIs, una vez que una de las 3 llamadas tenga éxito se abortarán las otras dos.

## **SETSELFALLERID [NúmeroPropio [HIDE]]**

*Descripción:* Define el valor de nuestro CallerID que se envía en todas las llamadas externas que hagamos en un script, a partir de esta instrucción, por una línea Digital (Básico o Primario), con los comandos CALL2EXTERIOR, CALL2EXTERIORBG o CALL2EXTENSION (en el caso de extensiones externas).

Si la llamada se realiza por una línea analógica esta instrucción no tiene ningún efecto.

Cuando realizamos una llamada por una línea digital, la operadora puede tener configurado por defecto, que se mande CallId Oculto, o que se mande uno de los números asignados a esa línea digital.

*NúmeroPropio* es el número de teléfono que queremos que le llegue al destino como número origen de la llamada, este *NúmeroPropio*, en principio, tiene que pertenecer al grupo del que forma parte el canal por el que hagamos la llamada, si no pertenece, nuestra operadora puede permitirnos que lo enviemos o puede no tener ningún efecto y la llamada se realice acorde a lo que tenga configurado la operadora por defecto.

*HIDE* indica que se envíe CallId oculto, en este caso también es necesario que el *NúmeroPropio* indicado pertenezca al grupo del que forma parte el canal por el que hagamos la llamada, si no pertenece no tiene ningún efecto y la llamada se realiza acorde a lo que tenga configurado la operadora por defecto.

Si no se indica ningún parámetro la llamada se realiza acorde a lo que tenga configurado la operadora por defecto.

Cuando se llama a otro fichero script con el comando CALL se transmite el valor definido con esta instrucción, e igual ocurre al volver al fichero desde el que ha sido llamado un script con el comando CALL.

Cuando se lanza un fichero script con el comando INVOKE no se transmite el valor definido con esta instrucción.

El valor de CallerID definido aquí, tiene prioridad sobre el valor que pudiera haberse definido en el fichero Sutil.ini para la línea que se utilice para realizar la llamada.

## **SETOFFHOOKTIMEOUT *Canal Time***

*Descripción:* Con este comando podemos indicarle a Sutil un *Time*, transcurrido el cual desde que lanzamos una llamada (con los comandos CALL2EXTERIOR o CALL2EXTERIORBG) por *Canal*, debe dar por exitosa la llamada, aunque no se reciba un evento de descolgado.

Esto puede ser útil para poder llamar por líneas que no tengan cambio de polaridad, como ocurre a veces en el caso de conectar líneas de Sutil a extensiones de una centralita.

También puede ser útil para poder oír mensajes que el operador nos reproduzca antes de descolgarnos.

Lo que podamos indicar con este comando no tiene ningún efecto en las posteriores llamadas que se hagan en el mismo script con los comandos CALL2EXTERIORISDN o CALLSIMULTANEUSBG.

*Salidas anómalas:* 4.- Número de canal no válido.  
(debe tener un valor entre 0 y 20).

## **GETOPENCHANNEL *Canal Tipo Línea***

*Descripción:* Busca en el sistema una línea externa o una extensión y la asocia a *Canal*.

*Tipo* puede valer:

1 .- Indica que se trata de una línea externa, en este caso *Línea* es el número de línea externa numerando primero los PRI, luego los BRI, luego las RTC y por último las VoIP.

2 .- Indica que se trata de una extensión del sistema y en este caso *Línea* es el número de extensión.

La línea o extensión debe estar "conectada" a sutil, es decir debe ser un canal abierto.

Si la línea o extensión estaba asociada a un canal de otro script se deshace la asociación y provoca un ONHOOK en dicho script.

*Salidas anómalas:* 4.- Línea o extensión inexistente.  
Tipo erróneo (no es ni 1 ni 2).  
5.- La línea o extensión no está "conectada" con el motor.  
7.- Canal en uso.



**CONNECT Canal1 Canal2 [Modo]**

*Descripción:* Conecta Canal1 con Canal2.

Modo puede valer:

0 .- Es el valor por defecto. Realiza una conexión bidireccional desconectandolos antes totalmente y parando ejecuciones previas en ambos canales de comandos PLAYFILE, PLAYTONE, ...

1 .- Realiza una conexión unidireccional del Canal1 al Canal2. Es decir el Canal2 puede escuchar al Canal1 pero no a la inversa.

2 .- Realiza una conexión unidireccional del Canal2 al Canal1. Es decir el Canal1 puede escuchar al Canal2 pero no a la inversa.

Si sobre Canal1 o Canal2 se está ejecutando una instrucción CALL2EXTERIORBG o CALLSIMULTANEOUSBG, no se comprueba el Modo y se realiza automáticamente una conexión en modo 1 si la instrucción de llamada saliente se ejecuta sobre el Canal1 y en modo 2 si se ejecuta la instrucción de llamada saliente sobre el Canal2. Es decir la conexión se hace de forma que el otro canal escuche al canal por el que se está ejecutando la instrucción CALL2EXTERIORBG o CALLSIMULTANEOUSBG.

*Salidas anómalas:* 4.- Canal no abierto o modo incorrecto.

**CONNECT2SPEAKER Canal**

*Descripción:* Une el canal especificado al primer sistema de megafonía.

*Salidas anómalas:* 4.- Canal no abierto.

**UNCONNECT Canal**

*Descripción:* Desune la unión que pudiera tener este canal con otro a través del comando CONNECT o con un salón con CONNECT2ROOM

*Salidas anómalas:* 4.- Canal no abierto.

**HOOK Canal [Causa]**

*Descripción:* Cuelga la línea indicada por canal, antes de colgar dicha línea, si el canal esta conectado con otro a través del comando CONNECT o con un salón con CONNECT2ROOM, lo desconecta, por lo que no es necesario realizar antes un UNCONNECT Canal.

Opcionalmente podemos enviar la causa por la que se cuelga la llamada, según la tabla de causas RDSI.

Este comando no provoca el salto a la etiqueta que pudiera estar configurado con un comando ONHOOK.

*Salidas anómalas:* 4.- Canal no abierto.

## **ONHOOK Canal [Etiqueta]**

*Descripción:* En el caso de que se corte la comunicación en el canal indicado por causas exteriores al script (es decir, en cualquier caso salvo que se ejecute un comando HOOK o INCOMINGCALL 1), salta a la etiqueta indicada. Si no se indica etiqueta, no se hace nada en caso de que se corte. Por defecto, las opciones son que se acaba la ejecución del script en caso de que se produzca un corte en el canal 0 y no se hace nada en el caso de que se produzca un corte en cualquiera de los otros canales.

## **CONNECT2ROOM Canal Room [Opcion [NVarRespuesta]]**

*Descripción:* Conecta el canal al salón especificado por *Room*. *Opcion* puede ser 0 ó 1. Por defecto es 0 y es una conexión bidireccional. 1 es una conexión en sólo escucha.

Si *Room* vale 0 conecta al primer salón disponible (sin ninguna conexión), y en la variable *NvarRespuesta* devuelve el número del salón al que se ha conectado, si no se ha indicado *NvarRespuesta* se devuelve en la variable 0. Si no hay ningún salón disponible retorna el código de error 5.

Exceptuando el caso anterior *Room* debe tener un valor entre 1 y 255.

Un canal solo se puede conectar a un salón.

Un salón admite como máximo 30 conexiones simultáneas.

*Salidas anómalas:*

4.- Número de canal no válido.

Canal no abierto.

Número de variable no válido (fuera del rango 1-39).

5.- Número de salón no válido(fuera del rango 1-255).

El salón no admite más conexiones.

No hay ningún salón disponible (si se ha indicado salón 0).

7.- Canal ya conectado a un salón.

## **GETINFOROOM NumSalon NumVar**

Disponible desde v.2.11.2.0

*Descripción:* Nos da información de cuantas de conexiones activas tiene el salón indicado por *NumSalon*.

*Salidas anómalas:*

4.- Número de variable no válido (fuera del rango 1-39).

5.- Número de salón no válido(fuera del rango 1-255).

## **GETINFOCANAL Canal NVarTipo NVarNum NVarDir**

*Descripción:* Lee la información interna sobre *Canal* y nos devuelve información sobre él.

Se puede utilizar sobre cualquier canal abierto, también se puede utilizar sobre el canal 0 antes de que la llamada sea descolgada con INCOMINGCALL, y en un canal sobre el que se este ejecutando un CALL2EXTENSIONBG.

*NVarTipo* es el número de variable donde nos devuelve el tipo:

- 2 - Usuario externo (origen en llamadas entrantes o destino en llamadas salientes).
- 3 - Usuario externo VoIP ("From" en llamadas entrantes o "To" en llamadas salientes).
- 4 - Extensión Interna.
- 5 - Extensión Externa VoIp.
- 6 - Extensión Externa Telefónica.
- 7 - Extensión Externa en otro Sutil.

*NVarNum* es el número de variable donde nos devuelve la dirección lógica, que según el tipo es:

En los casos 4, 5, 6 y 7: el número de extensión lógica.

En el caso 2: el número de teléfono.

En el caso 3: Usuario Sip en llamadas entrantes, o destino en llamadas salientes.

*NVarDir* es el número de variable donde nos devuelve la dirección física, que según el tipo es:

En el caso 2: el número de línea física por la que ha entrado o salido la llamada.

En el caso 3: el número de línea externa VoIP.

En el caso 4: el número de extensión física.

En el caso 5 (Protocolo SIP): el nombre de usuario SIP asociado.

En el caso 5 (Protocolo H323): la dirección IP asociada.

En el caso 6: el número de teléfono asociado a la extensión.

En el caso 7: la dirección IP asociada al host.

*Salidas anómalas:*

4.- Número de canal no válido.

Canal no abierto.

Número de variable no válido (fuera del rango 1-39).

---

**GETINFOGROUP *NExt* *NVarTotal* *NVarAuten* *NvarTipo* [ *NvarActi*  
[ *NvarDispo* [ *NvarCola* ] ] ]**

*Descripción:* Nos da información sobre el número total de extensiones que contiene un grupo, cuantas están autenticadas y alguna otra información.

*NExt* contiene el número de extensión o grupo del que queremos obtener la información.

*NVarTotal* es el número de variable que contiene el número total de extensiones que contiene el grupo, incluyendo, si las hay, las que están en otro Sutil.

Si se ha indicado una extensión esta variable contendrá siempre un uno.

*NVarAuten* es el número de variable que contiene el número de extensiones autenticadas en el grupo en ese momento, si el grupo incluye extensiones de otros sutiles siempre se entiende que están autenticadas.

Si se ha indicado una extensión contendrá 1 si la extensión esta autenticada y 0 si no lo esta.

*NVarTipo* es el número de variable donde nos indica 1 si la consulta se ha hecho sobre un grupo o 0 si la consulta se ha hecho sobre una extensión.

*NVarActi* es el número de variable que contiene el número de extensiones activas en el grupo en ese momento, si el grupo incluye extensiones de otros Sutil siempre se entiende que están activas.

Si se ha indicado una extensión contendrá 1 si la extensión esta activa y 0 si no lo esta.

*NvarDispo* es el número de variable que contiene el número de extensiones que están disponibles para ser llamadas en el grupo en ese momento.

Si se ha indicado una extensión contendrá 1 si la extensión esta disponible y 0 si no lo esta. En este caso, para decidir si la extensión esta disponible o no, se tiene en cuenta los derechos que estén activos en ese momento para llamar a extensiones, que previamente se hayan podido definir con el comando SETCALLRIGHTS.

Si se ha indicado un grupo se considera que una extensión esta disponible para ser llamada cuando:

Si es una extensión de cualquier tipo está activa y colgada.

Si es una softExtension está activa y no está conectada con nadie (esté colgada o descolgada).

*NvarCola* es el número de variable que contiene el número de llamadas en cola en ese grupo o extensión.

*Salidas anómalas:* 4.- Variable, grupo o extensión inexistente.

## RECEIVEFAX *Canal NvarRet Fichero*

*Descripción:* Se recibe un Fax por un canal que ya debe estar abierto. El Fax lo guarda en *Fichero* en un formato propio, normalmente al fichero se le pone de extensión .eif .

*NvarRet* contiene una X mientras se esta recibiendo el FAX, una vez finaliza la recepción del FAX contiene el resultado, que puede ser uno de los siguientes:

- 0 - Enviado correctamente.
- 1 - La tarjeta no responde.
- 2 - El origen no es una línea de FAX.
- 3 - Conexión remota fallida (Tal vez mala calidad en la línea ).
- 4 - Fallo en la recepción.
- 5 - Error de I/O. Formato de fichero erróneo, nombre de fichero no válido, ...
- 6 - Fallo en la fase de confirmación.
- 7 - Fallo en la recepción de multi-paginas.

*Fichero* es el nombre del fichero donde se guarda el fax recibido, puede incluir una ruta, se toma como directorio de partida para almacenar el fichero al directorio donde se ejecuta Sutil ( Normalmente C:\Sutil ).

Salidas anómalas:

- 4.- Número de canal no válido.  
Canal no abierto.  
Número de variable no válido (fuera del rango 1-39).
- 5.- No hay recursos de Fax disponibles.
- 7.- Ya se esta recibiendo o enviando un FAX por ese canal.

Ejemplo:

El siguiente script descuelga una llamada y recibe un FAX.

El FAX lo almacena en un fichero con nombre %nums, es decir con el nombre igual al número de teléfono desde donde ha sido enviado.

```
INCOMINGCALL 0

RECEIVEFAX 0 0 %nums.eif      ; Recibir el fax
IFERR 4 5 7                  ; Si no puede empezar a recibir....
    EXIT                      ; ...terminamos.
ENDIFERR

:ESPERAR_FIN                 ; Ciclo de comprobacion de la recepción
    IF %v0 != X                ; Comprobar si ha terminado ya.
        HOOK 0                 ; Si terminó ya, colgamos la linea...
        EXIT                   ; ... y terminamos.
    ENDIF
GOTO ESPERAR_FIN
```

**SENDER Canal NVarRet Fichero NumFax [NombreDestino]**

*Descripción:* Envía un Fax por un canal que ya debe estar abierto.

*NVarRet* contiene una X mientras se esta enviando el FAX, una vez finaliza el envío del FAX contienen el resultado, que puede ser uno de los siguientes:

- 0 - Enviado correctamente.
- 1 - La tarjeta no responde.
- 2 - El destino no tiene papel o no es una línea de FAX.
- 3 - Conexión remota fallida (Tal vez mala calidad en la línea ).
- 4 - Fallo en la transmisión.
- 5 - Error de I/O. Formato de fichero erroneo, el fichero no existe, ...
- 6 - Fallo en la fase de confirmación.
- 7 - Fallo en la transmisión de multi-paginas.

*Fichero* contiene una lista de los ficheros que hay que enviar, cada uno de estos ficheros se envían como una página, estos ficheros deben estar en un formato especial propio ( normalmente son ficheros con extensión .eif ).

*NumFax* indica cuantos ficheros de los que se indican en *Fichero* hay que enviar.

Si se indica *NombreDestino* aparecera en el monitor de estado.

Salidas anómalas:

- 4.- Número de canal no válido.  
Canal no abierto.  
Número de variable no válido (fuera del rango 1-39).
- 5.- No hay recursos de Fax disponibles.
- 7.- Ya se esta recibiendo o enviando un FAX por ese canal.

Ejemplo:

El siguiente script llama al nº de teléfono 952650460 y si tiene éxito la llamada envía las 3 páginas que se indican en el fichero PRESUPUESTO

El contenido del fichero PRESUPUESTO es:

```
PAGINA_1.fax
PAGINA_2.fax
PAGINA_3.fax
```

Es importante que cada una de estas líneas acabe con un retorno de carro (car. 0Dh) y un avance de línea (car. 0Ah).

```
CALL2EXTERIOR 1 952650460 30
IFERR 1 2 5 7
EXIT
ENDIFERR
```

```
SENDER 1 0 PRESUPUESTO 3
```

```

IFERR 4 5 7
    EXIT
ENDIFERR

:ESPERAR_FIN
    IF %v0 != X
        EXIT
    ENDIF
    GOTO ESPERAR_FIN

```

## SETLOCALIDFAX [*Identificación*]

*Descripción:* Define el valor del *Identificador Local* con el que se recibirán y enviarán los Fax desde un script, a partir de esta instrucción, con los comandos RECEIVEFAX y SENDFAX.

El valor que se defina con este comando para el *Identificador Local*, tiene prioridad sobre lo que se haya podido definir para dicho valor en el fichero *Sutil.ini*.

Cuando se llama a otro fichero script con el comando CALL se transmite el valor definido con esta instrucción, e igual ocurre al volver al fichero desde el que ha sido llamado un script con el comando CALL.

Cuando se lanza un fichero script con el comando INVOKE no se transmite el valor definido con esta instrucción.

## CALL2VOIP *Canal UsuarioSip@DireccionIP Timeout*

*Descripción:* Hace una llamada VoIP al *UsuarioSip@DireccionIP* indicado y la asocia al canal indicado. No hace llamadas a extensiones, sino a usuarios que se registran en un servidor SIP.

*Timeout* indica el máximo tiempo en segundos a esperar, este valor debe estar comprendido normalmente entre 20 y 40.

Salidas anómalas:

- 1.- Timeout.
- 2.- Comunicando.
- 5.- No hay disponibles recursos VoIP para llamar.
- 7.- Canal en uso.

Ejemplo:

```

CALL2VOIP 0 pepito@217.11.100.10:5060 20
CALL2VOIP 0 216@217.11.100.10 20

```

## **SENDUSERDATA Canal Texto**

*Descripción:* Envía un mensaje de usuario por una línea RDSI.

Este comando debe ejecutarse sobre un *Canal* por el que exista una llamada y se haya recibido/enviado como mínimo una alerta, no es necesario que el *Canal* este descolgado, es decir un *Canal* en el que se haya realizado una llamada y que este sonando o ya nos hayan respondido o por el que hayamos recibido una llamada, hayamos descolgado o no.

El mensaje puede tener de tamaño como máximo 256 caracteres.

*Salidas anómalas:*       4.- Número de canal no válido.  
Canal no abierto.

## **READUSERDATA Canal NVar**

*Descripción:* Si se recibe un mensaje de usuario por una línea RDSI se almacena en un buffer, este buffer admite como máximo 10 mensajes, si este buffer ya está lleno y se reciben más mensajes se descartan.

Este comando nos devuelve en la variable indicada el primer mensaje (más antiguo) del buffer y lo borra de dicho buffer.

Este comando debe ejecutarse sobre un *Canal* por el que exista una llamada y se haya recibido/enviado como mínimo una alerta, no es necesario que el *Canal* este descolgado, es decir un *Canal* en el que se haya realizado una llamada y que este sonando o ya nos hayan respondido o por el que hayamos recibido una llamada, hayamos descolgado o no.

El mensaje puede tener de tamaño como máximo 256 caracteres.

*Salidas anómalas:*       4.- Número de canal no válido.  
Canal no abierto.  
Número de variable no válido.



## 2.6.- Utilización de recursos de voz.

Estos comandos utilizan recursos de voz para reproducir o grabar voz humana a través de los interfaces telefónicos disponibles. Siempre hay que especificarles el canal a través del que actúan. Normalmente este canal será 0, que siempre es el canal de la llamada entrante. Solamente se utilizarán otros canales si el script inicia otras llamadas simultáneas. Los números de canal pueden tomar un valor de 0 a 9.

Los comandos PLAYTONE y BEEP no consumen ningún recurso de voz, aunque se han incluido en este apartado por similitud con los otros comandos.

### **PLAYFILE Canal Fichero [ Opcion [ NivelTraza ] ]**

*Descripción:* Reproduce en *Canal* el *Fichero* wav indicado.

La reproducción en background (*Opcion* 3, 4, 5 y 6) se para por otro comando PLAYFILE, PLAYNUMBER, PLAYTIME, PLAYTONE, INPUT (Solo en los casos de *Opcion* 3 y 4, no en los casos 5 y 6), CONNECT, CONNECT2ROOM, CONNECT2SPEAKER o RECORDFILE (No se para por un RECORDFILEBG).

*Fichero* puede llevar incluida o no la extensión .wav .

*Fichero* se busca por defecto en *Sutil\WavFiles\* , y puede incluir un camino hasta el fichero, en cualquier caso se parte del directorio *Sutil\WavFiles\* , salvo que *Fichero* comience por:

“\” en este caso se parte del directorio raíz de la unidad donde se este ejecutando Sutil.exe.

“letra de unidad:” en este caso, se indique explícitamente o no, siempre se parte del directorio raíz de la unidad indicada.

“\\Nombre\_ordenador\Recurso\_compartido” o

“\\Dir\_IP\Recurso\_compartido” en estos dos casos, se indique explícitamente o no, siempre se parte del directorio raíz del Recurso\_compartido.

En todos estos casos podemos utilizar “\” o “/” indistintamente.

*Opcion*, puede valer de 0 a 4. 0 es el valor por defecto.

- 0 - El mensaje parará al recibir un tono DTMF o al acabar.
- 1 - El mensaje parará al acabar, pero no si se reciben tonos DTMF.
- 2 - El mensaje se reproduce continuamente hasta que se reciba un DTMF.
- 3 - El mensaje se reproduce en background.
- 4 - El mensaje se reproduce en background continuamente.
- 5 - El mensaje se reproduce en background, parará al recibir un tono DTMF o al acabar.
- 6 - El mensaje se reproduce en background continuamente, parará al recibir un tono DTMF.

Si la reproducción se para por recibir un tono DTMF, se abre un buffer (si no estaba ya abierto) y se almacena en él, el DTMF recibido para que pueda ser recogido después por un comando INPUT.

*NivelTraza*, puede valer de 0 a 5, nos permite modificar el nivel con que se generan las 2 siguientes trazas:

- "No existe el fichero que se quiere reproducir" (108.09.42015.4).
- "El canal no esta abierto o no hay canales disponibles" (108.09.42016.3).

Si no ponemos nada no se modifica el nivel con que se generan estas 2 trazas.

Mientras se esta ejecutando este comando se comprueba si se produce alguna interrupción, si se produce un cuelgue del canal por el que esta reproduciendo este comando y está deshabilitado el ONHOOK este comando da error 4 en ese momento (ademas de generar una traza 108.09.42016.3). Por lo que se recomienda tener activo el ONHOOK adecuado para el caso de que se produzca el cuelgue del canal durante la ejecución de este comando.

- Salidas anómalas:*
- 4.- Fichero inexistente o canal no abierto.
  - 5.- Recurso de voz no disponible.

## **PLAYTIME Canal FormatoFecha FormatoHora Cadena [Opcion [Idioma]]**

*Descripción:* Reproduce una fecha y/o hora a través de *Canal*.

*FormatoFecha* indica en la forma *abcde* como reproducir la fecha:

- a: 0 No reproducir fecha.  
1 Reproducir fecha.
- b: 0 No reproducir artículo previo.  
1 Reproducir artículo previo.
- c: 0 Sólo reproducir año si no es el actual.  
1 Reproducir año en todos los casos.
- d: 0 No reproducir día de la semana.  
1 Reproducir día de la semana.
- e: 0 Reproducir fecha numérica normal.  
1 Si la fecha es la de ayer, hoy o mañana, reproducir la palabra "ayer", "hoy" o "mañana" respectivamente.

*FormatoHora* indica en la forma *abcde* como reproducir la hora:

- a: 0 No reproducir hora.  
1 Reproducir hora.
- b: 0 No reproducir artículo previo.  
1 Reproducir artículo previo.
- c: 0 Reproducir en formato 24 horas.  
1 Reproducir en formato am/pm.
- d: 0 No reproducir sustantivos adicionales.  
1 Reproducir sustantivos "horas" y "minutos".

- b: 0 No reproducir segundos.  
1 Reproducir segundos.

*Cadena* es la cadena de fecha y/o hora. Puede tener uno de estos formatos:

hh:mm  
hh:mm:ss  
aaaa/mm/dd  
aaaa/mm/dd hh:mm  
aaaa/mm/dd hh:mm:ss  
dd/mm/aa  
dd/mm/aa hh:mm  
dd/mm/aa hh:mm:ss

*Opcion*, puede valer 0, 1 o 2. 0 es el valor por defecto.

- 0 - El mensaje parará al recibir un tono DTMF o al acabar.  
1 - El mensaje parará al acabar, pero no si se reciben tonos DTMF.  
2 - El mensaje se reproduce continuamente hasta que se reciba un DTMF.

*Idioma* es el código de idioma. En el *Manual del Administrador de Sutil*, en el capítulo de Datos Técnicos podemos consultar la tabla de Idiomas. Si no se especifica se utiliza el castellano ( 0 ).

*Salidas anómalas:* 4.- Canal no abierto.  
5.- Recurso de voz no disponible.

## **PLAYMONEY Canal Cadena [Opcion [Moneda [Idioma]]]**

*Descripción:* Reproduce un importe en la *Moneda* e *Idioma* indicado a través de *Canal*.

*Cadena* es el importe a reproducir, debe estar comprendido entre 0 y 999.999.999 hasta con 2 decimales (separados por un punto ).

*Opcion*, puede valer de 0 a 2. 0 es el valor por defecto.

- 0 - El mensaje parará al recibir un tono DTMF o al acabar.  
1 - El mensaje parará al acabar, pero no si se reciben tonos DTMF.  
2 - El mensaje se reproduce continuamente hasta que se reciba un DTMF.

*Moneda* es el código de la moneda. En el *Manual del Administrador de Sutil*, en el capítulo de Datos Técnicos podemos consultar la tabla de Monedas. Si no se especifica se utiliza el Euro ( 0 ).

*Idioma* es el código de idioma. En el *Manual del Administrador de Sutil*, en el capítulo de Datos Técnicos podemos consultar la tabla de Idiomas. Si no se especifica se utiliza el castellano ( 0 ).

*Salidas anómalas:* 4.- Canal no abierto.  
5.- Recurso de voz no disponible

## **PLAYNUMBER Canal Cadena Genero [Opcion [Idioma]]**

*Descripción:* Reproduce una cadena numérica a través de *Canal*.

*Cadena* es el número a reproducir.

*Genero* puede valer de 0 a 4.

- 0 - reproduce un número entero en masculino ("uno").
- 1 - en neutro ("un").
- 2 - en femenino ("una").
- 4 - reproduce el número en masculino cifra a cifra.
- 6 - reproduce el número en femenino cifra a cifra.

En los casos 0, 1 y 2 el número debe estar comprendido entre 0 y 999.999.999 .

*Opcion*, puede valer 0, 1 o 2. 0 es el valor por defecto.

- 0 - El mensaje parará al recibir un tono DTMF o al acabar.
- 1 - El mensaje parará al acabar, pero no si se reciben tonos DTMF.
- 2 - El mensaje se reproduce continuamente hasta que se reciba un DTMF.

*Idioma* es el código de idioma. En el *Manual del Administrador de Sutil*, en el capítulo de Datos Técnicos podemos consultar la tabla de Idiomas. Si no se especifica se utiliza el castellano ( 0 ).

*Salidas anómalas:*

- 4.- Canal no abierto.
- 5.- Recurso de voz no disponible

## **PLAYORDINAL Canal Cadena Genero [Opcion [Idioma]]**

*Descripción:* Reproduce una cadena numérica a través de *Canal* en ordinal.

*Cadena* es el número a reproducir, debe estar comprendido entre 1 y 40 .

*Genero* puede valer de 0 a 2:

- 0 reproduce un número entero en masculino ("primero").
- 1 en neutro ("primer").
- 2 en femenino ("primera").

*Opcion*, puede valer 0, 1 o 2. 0 es el valor por defecto.

- 0 El mensaje parará al recibir un tono DTMF o al acabar.
- 1 El mensaje parará al acabar, pero no si se reciben tonos DTMF.
- 2 El mensaje se reproduce continuamente hasta que se reciba un DTMF.

*Idioma* es el código de idioma. En el *Manual del Administrador de Sutil*, en el capítulo de Datos Técnicos podemos consultar la tabla de Idiomas. Si no se especifica se utiliza el castellano ( 0 ).

*Salidas anómalas:*

- 4.- Canal no abierto.
- 5.- Recurso de voz no disponible

## **PLAYTONE Canal [TipoTono [ModoParada]]**

*Descripción:* Reproduce un tono determinado en background. Este play no consume recurso de voz alguno.

*TipoTono*, puede valer de 0 a 4. Si no se especifica ninguno, anula la ejecución de un PLAYTONE previo.

0 - es un tono de llamada sonando (RING)

1 - es un tono de invitación a marcar.

2 - es un tono de comunicando.

3 - es música en espera electrónica prefijada.

4 - es música en espera proveniente del conector jack de entrada de sonido o de un fichero wav, si así se ha indicado en el fichero Sutil.ini.

5 - es un tono de Alarma.

*ModoParada*, puede valer 0 ó 1. 0 es el valor por defecto.

0 - El tono se reproducirá hasta que se produzca una conexión del canal, o se inicie la reproducción de otro mensaje o tono.

1 - El tono se reproducirá hasta que se ejecute un comando PLAYTONE Canal.

*Salidas anómalas:* 4.- Canal no abierto.

5.- Recurso de voz no disponible.

## **PLAYDTMF Canal Cadena**

*Descripción:* Reproduce los tonos DTMF contenidos en *Cadena* por el canal especificado.

*Cadena* puede contener números o uno de estos tres símbolos: # \* ,

La coma significa una pausa de un segundo.

*Salidas anómalas:* 4.- Canal no abierto.

5.- Recurso de voz no disponible.

Ejemplo:

En este ejemplo Sutil es capaz de interoperar con otro sistema telefónico sin necesidad de que un operador humano tenga que elegir opciones o introducir códigos.

El siguiente ejemplo descuelga una llamada entrante, le pone tono de llamando, llama a un número de teléfono donde nos atiende un sistema automático que primero nos da a elegir entre varias opciones, elegimos la segunda y después nos pide un código que introducimos el "1234", y por ultimo unimos la llamada entrante con la llamada saliente, se han colocado algunas pausas para adaptarse al sistema automático que nos atiende.

```
INCOMINGCALL 0
```

```
PLAYTONE 0 0
```

```
CALL2EXTERIOR 1 902226022 30
```

```
IFERR 0
  PLAYDTMF 1 , , 2 , , 1234
  PAUSE 2
  CONNECT 0 1
  EXIT
ENDIFERR
PLAYTONE 0 2
```

## **BEEP Canal**

*Descripción:* Reproduce un pitido en el canal indicado. Es útil, para indicar mensajes del tipo “diga nosequé después de la señal”.

Si el canal indicado esta unido con otro u otros canales, el pitido solo se escucha en el canal indicado, no se escucha en el otro u otros canales.

*Salidas anómalas:* 5.- Recurso de voz no disponible.

## **RECORDFILE Canal Fichero Maxtiempo Opción [TiempoSilencio [Attach]]**

*Descripción:* Graba un mensaje en el fichero indicado por *Fichero*. Se le puede indicar que termine de grabar por varios motivos.

Este comando solo graba el *Canal* indicado, no los posibles canales conectados con él. Para grabar también estos canales, debemos utilizar el comando RECORDFILEBG con los parámetros correspondientes.

*Opción* puede valer:

- 0 - Es la opción por defecto. Acaba de grabar cuando finaliza el tiempo indicado por *MaxTiempo* (en seg.).
- 1 - Además de acabar como la opción 0, también se acaba de grabar si se recibe un tono DTMF.
- 2 - Además de acabar como la opción 0, también se acaba de grabar si hay un silencio mayor a *TiempoSilencio* (en seg.), si no se indica este valor se toma por omisión 0 seg..
- 3 - Acaba por cualquiera de los 3 motivos anteriores.

Si la grabación se para por recibir un tono DTMF, se abre un buffer (si no estaba ya abierto) y se almacena en él el DTMF recibido, para que pueda ser recogido después por un comando INPUT.

*Fichero* se genera por defecto en *Sutil\WavFiles\* , y puede incluir un camino hasta el fichero, en cualquier caso se parte del directorio *Sutil\WavFiles\* , salvo que *Fichero* comience por:

“\” en este caso se parte del directorio raíz de la unidad donde se este ejecutando Sutil.exe.

“letra de unidad:” en este caso, se indique explícitamente o no, siempre se parte del directorio raíz de la unidad indicada.

“\\Nombre\_ordenador\Recurso\_compartido” o

“\\Dir\_IP\Recurso\_compartido” en estos dos casos, se indique explícitamente o no, siempre se parte del directorio raíz del Recurso\_compartido.

En todos estos casos podemos utilizar “\” o “/” indistintamente.

*Attach* puede valer (0 es el valor por defecto):

0 .- Si *Fichero* ya existe se borra antes de empezar a grabar.

1 .- Si *Fichero* ya existe la grabación se añade al fichero existente.

*Salidas anómalas:* 4.- Nombre de fichero no válido o canal no abierto.

5.- Recurso de voz no disponible.

## **RECORDFILEBG Canal Fichero Maxtiempo Opción [TiempoSilencio [Attach]]**

*Descripción:* Comienza a grabar el *Canal*, y todos los canales conectados a *Canal*, ya sea mediante el comando CONNECT o CONNECT2ROOM, en el fichero indicado por *Fichero*.

Opcionalmente, podemos grabar solo el canal indicado, y no los canales conectados a él.

La ejecución del script continúa.

Este comando no funciona si se utiliza cuando la llamada asociada a *Canal* se esta grabando.

Cuando finaliza el tiempo indicado por *MaxTiempo* (en seg.), si el script continúa ejecutándose, se deja de grabar.

Si mientras esta grabándose un canal, por la ejecución de este comando, se ejecuta un EXIT del script, si el canal esta conectado a otro canal se sigue grabando hasta que el canal se cierre, independientemente de que haya transcurrido o no el tiempo indicado por *MaxTiempo* (en seg.).

*Opción* puede constar de dos caracteres:

El primer carácter indica:

- 0 (o vacío) - Se graba el *Canal* y todos los canales conectados.
- 1 - Se graba solo el *Canal* indicado.

El Segundo carácter indica:

- 0 - Es la opción por defecto. Acaba de grabar cuando finaliza el tiempo indicado por *MaxTiempo* (en seg.).
- 1 - Además de acabar como la opción 0, también se acaba de grabar si se recibe un tono DTMF.
- 2 - Además de acabar como la opción 0, también se acaba de grabar si hay un silencio mayor a *TiempoSilencio*, si no se indica este valor se toma por omisión 0 seg..
- 3 - Acaba por cualquiera de los 3 motivos anteriores.

Si la reproducción se para por recibir un tono DTMF, si en ese momento también hay en marcha un comando INPUT, dicho comando INPUT recoge el DTMF pulsado, en caso contrario el DTMF recibido NO se almacena en un buffer para que pueda ser recogido después por un comando INPUT.

*Fichero* se genera por defecto en *Sutil\WavFiles\*, y puede incluir un camino hasta el fichero, en cualquier caso se parte del directorio *Sutil\WavFiles\*, salvo que *Fichero* comience por:



“\” en este caso se parte del directorio raíz de la unidad donde se este ejecutando Sutil.exe.

“letra de unidad:” en este caso, se indique explícitamente o no, siempre se parte del directorio raíz de la unidad indicada.

“\\Nombre\_ordenador\Recurso\_compartido” o

“\\Dir\_IP\Recurso\_compartido” en estos dos casos, se indique explícitamente o no, siempre se parte del directorio raíz del Recurso\_compartido.

En todos estos casos podemos utilizar “\” o “/” indistintamente.

*Attach* puede valer (0 es el valor por defecto):

0 - Si *Fichero* ya existe se borra antes de empezar a grabar.

1 - Si *Fichero* ya existe la grabación se añade al fichero existente.

<b>¡ ATENCIÓN !</b>	<p>Si indicamos un Canal del que ya se esta haciendo una grabación, debido a que esta configurado grabar llamadas Entrantes o Salientes, según corresponda, este comando no graba nada, pero si indicamos el otro canal con el que este unido dicho canal, si se graba, ejemplo:</p> <p>Si en configuración esta indicado que se grabe las llamadas entrantes, y tenemos una llamada entrante en el canal 0, hacer el RECORDFILEBG del canal 1 si funciona, pero no funciona si lo hacemos del canal 0.</p>
---------------------	---

*Salidas anómalas:* 4.- Nombre de fichero no válido o canal no abierto.  
5.- Recurso de voz no disponible.

## **INPUT Canal NVar Ncaract Timeout [CaracteresFin [CaractAdmitidos]]**

*Descripción:* Recibe una cadena de tonos DTMF en *Canal* y la introduce en la variable *Nvar*. También detecta la pulsación de la tecla R, en cuyo caso devuelve una R. A partir de la ejecución del primer comando INPUT en un canal o del corte de un comando PLAYFILE, RECORDFILE o RECORDFILEBG por la recepción de un tono DTMF, permanecerá abierto un buffer donde se almacenaran todos los tonos DTMF que se tecleen entre INPUTs. En caso de que se quieran borrar, habrá que utilizar el comando CLEARINPUT.

*Ncaract* es la máxima longitud. Cuando se han entrado estos caracteres, acaba.

*Timeout* es el máximo tiempo en segundos que espera sin pulsar ninguna tecla en el teléfono. La pulsación de la entrada completa puede tardar más tiempo.

*CaracteresFin* es una cadena opcional que contiene unos caracteres DTMF que provocan la finalización de la entrada. El carácter que haya provocado la salida no se incorpora a la cadena leída.

*CaractAdmitidos* es una cadena opcional que contiene los caracteres que serán admitidos. En el caso de existir esta cadena, cualquier carácter que no pertenezca a ella será descartado.

*Salidas anómalas:*

- 1.- Timeout.
- 4.- Canal no abierto o variable inexistente.
- 5.- Recurso de voz no disponible.

### **INPUTR Canal Timeout**

*Descripción:* Detecta la pulsación de la tecla R, en cuyo caso devuelve como retorno 0. No utiliza recurso de voz.

*Timeout* es el máximo tiempo en segundos que espera sin pulsar la tecla R en el teléfono.

*Salidas anómalas:*

- 1.- Timeout sin detectar R.
- 4.- Canal no abierto o variable inexistente.

### **CLEARINPUT Canal**

*Descripción:* Elimina todas las pulsaciones DTMF en el buffer de entrada del canal, eliminando la recepción de tonos DTMF en ese canal a partir de ese momento. Este buffer sólo existe si previamente se ha ejecutado INPUT en ese canal.

*Salidas anómalas:*

- 4.- Canal no abierto.

### **CHECKVOICE Canal Nvar TiempoSilencio**

*Descripción:* Comprueba si en el otro extremo de *Canal* hay Voz, Fax o silencio.

Este comando debe ejecutarse sobre un *Canal* que ya este abierto, es decir en el que se haya realizado una llamada y nos hayan respondido o por el que hayamos recibido una llamada y ya hayamos descolgado.

Este comando se utiliza normalmente y es útil en llamadas salientes, para descartar los faxes antes de pasar la llamada a una operadora. En llamadas entrantes normalmente no es útil, ya que el llamante (sea una persona o fax), no hace nada hasta que nosotros no decimos algo.

Nvar nos devuelve el tipo de interlocutor:

- 1 – Voz.
- 2 – Fax/modem.
- 3 – Silencio.

La Voz se reconoce en poco tiempo, 1 ó 2 segundos.

El Fax se tarda en reconocer entre 5 y 10 seg. aproximadamente.

Se reconoce Silencio cuando el tiempo de silencio continuo alcanza el valor indicado en *TimeSilence*, este valor debe ser inferior a 30 segundos, lo normal es indicar 2 ó 3 segundos.

Si en 30 segundos no se reconoce nada el comando acaba y devuelve Timeout.

- Salidas anómalas:*
- 1.- Timeout sin detectar nada.
  - 4.- Número de canal no válido.  
Canal no abierto.  
Número de variable no válido.
  - 5.- Recurso de voz no disponible.

## 2.7.- Gestión de extensiones y grupos.

Una extensión dinámica puede estar asignada a una extensión física, a un número de teléfono externo o a un fichero script. Esta asignación se puede controlar a través del comando SETEXTENSION. Cuando en los comandos figura un argumento de Número de extensión se refiere siempre al número de extensión dinámica, por lo que a veces una llamada a una extensión puede motivar una llamada al exterior por ejemplo.

En el caso del comando DISPLAYLCD, evidentemente la extensión ha de estar asignada a una extensión física local.

### SETEXTENSION *NumExt* *Tipo* [*Asignación*] [*Password*]

*Descripción:* Asigna la extensión virtual *NumExt* o la activa / desactiva.

*Tipo* puede valer:

0 - Se borra cualquier asignación de la extensión.

1 - Se le asigna el número de extensión física contenida en *Asignación*, esta asignación solo se realiza si la extensión lógica no esta ya autenticada ( en este caso se devuelve un error 7 ) y no hay ninguna extensión autenticada en la extensión física ( en este caso se devuelve un error 5 ).

2 - Se le asigna el número de teléfono contenido en *Asignación*, esta asignación solo se realiza si la extensión lógica no esta ya autenticada ( en este caso se devuelve un error 7 ) y no hay ninguna extensión autenticada en el número de teléfono ( en este caso se devuelve un error 5 ).

Si da ambas condiciones, es decir la extensión lógica indicada en *NumExt* ya está autenticada y el número de teléfono indicado en *Asignación* ya corresponde a una extensión autenticada, se devuelve un error 5.

3 - Se le asigna el script contenido en *Asignación*, esta asignación solo se realiza si la extensión lógica no esta ya autenticada ( en este caso se devuelve un error 7 ).

4 - (Protocolo SIP) Se le asigna el nombre de usuario SIP contenido en *Asignación*, esta asignación solo se realiza si la extensión lógica no esta ya autenticada ( en este caso se devuelve un error 7 ) y no hay ninguna extensión autenticada con ese nombre de usuario SIP ( en este caso se devuelve un error 5 ).

4 - (Protocolo H323) Se le asigna la dirección Ip contenida en *Asignación*, esta asignación solo se realiza si la extensión lógica no esta ya autenticada ( en este caso se devuelve un error 7 ) y no hay ninguna extensión autenticada en la dirección Ip ( en este caso se devuelve un error 5 ).

5 - Se le asigna el número de Servidor de enrutamiento contenido en *Asignación*, esta asignación solo se realiza si la extensión lógica no esta ya autenticada ( en este caso se devuelve un error 7 ) y el valor del

servidor de enrutamiento es válido ( en este caso se devuelve un error 4 ).

10 - Coloca la extensión en situación activa.

11 - Coloca la extensión en situación desactiva.

12 - Coloca la extensión en situación Trabajo entre llamadas (Back Office).

20 - Se borra cualquier asignación de la extensión si la tuviera y se le asigna el valor por defecto.

21 - Siempre que el valor de *Asignación* sea válido, se le asigna el número de extensión física contenida en *Asignación*.

22 - Siempre que el valor de *Asignación* sea válido, se le asigna el número de teléfono contenido en *Asignación*.

23 - Siempre que el valor de *Asignación* sea válido, se le asigna el script contenido en *Asignación*.

24 - (Protocolo H323) Siempre que el valor de *Asignación* sea válido, se le asigna la dirección Ip contenida en *Asignación*.

24 - (Protocolo SIP) Siempre que el valor de *Asignación* sea válido, se le asigna el nombre de usuario SIP contenido en *Asignación*.

25 - Siempre que el valor de *Asignación* sea válido, se le asigna el número de Servidor de enrutamiento contenido en *Asignación*.

Si indicamos el parámetro *Password* se comprueba, si no es correcto no se realiza el proceso de asignación y se devuelve el error 6.

Si no lo indicamos no se comprueba, aunque la extensión tenga asignado un password, y se realiza el proceso de asignación.

Si hemos indicado el tipo 0 o 20, no se debe indicar el parámetro *Asignación*, y en caso de que queramos indicar el *Password* lo pondríamos como tercer parámetro.

#### *Salidas anómalas:*

4.- Extensión, usuario o servidor de enrutamiento inexistente.

*%suberr:*

1.- Extensión física inexistente.

2.- Extensión lógica inexistente.

3.- Extensión no autenticada.

4.- Usuario no encontrado en la lista de usuarios.

5.- Servidor de enrutamiento no válido.

5.- Se ha intentado autenticar una extensión lógica en un emplazamiento donde ya esta autenticada otra extensión lógica.

6.- Password incorrecto.

7.- Se ha intentado autenticar una extensión lógica, que ya esta autenticada en otro emplazamiento.

## **SETCALLRIGHTS *Rights***

*Descripción:* Normalmente cuando desde un script llamamos a una extensión que no esta activa, con los comandos CALL2EXTENSION o CALL2EXTENSIONBG,

estos comandos nos lo indican y no llaman a la extensión (no la hacen sonar), con este comando podemos hacer que si se llame a la extensión también en los casos en que este en estado *Trabajando entre llamadas* o *Inactivo*.

A partir de que ejecutamos este comando en un script todos los CALL2EXTENSION y CALL2EXTENSIONBG que ejecutemos, se ejecutarán con los *Rights* que se hayan indicado.

Solo se aplica cuando se llama a una extensión y no tiene ningún efecto cuando se llama a un grupo.

*Rights* son los niveles de derechos del llamante.

0 - No se puede llamar a extensiones en estado *Inactivo* (\*11) o *Trabajo entre llamadas* (\*10). Derechos normales de las extensiones *Usuario*.

1 - No se puede llamar a extensiones en estado *Inactivo* (\*11) pero si en estado *Trabajo entre llamadas* (\*10).

2 - Si se puede llamar a extensiones en estado *Inactivo* (\*11) y *Trabajo entre llamadas* (\*10). Derechos normales de las extensiones *Administrador*.

## **DISPLAYLCD NExt MensajeLCD**

(Será implementado próximamente)

*Descripción:* Envía un mensaje a una extensión, para ello deberá sonar un ring en dicha extensión.

*NExt* es el número de extensión a donde enviar el mensaje.

*MensajeLCD* es el mensaje a mostrar.

*Salidas anómalas:*

- 1.- Timeout.
- 2.- Comunicando.
- 4.- Extensión inexistente.

## **TRANSFER2EXTENSION NExt Timeout MensajeVoz [MensajeLCD]**

*Descripción:* Transfiere la llamada a una extensión. Si lo consigue finaliza la ejecución del script. Si la extensión no responde o comunica, el sistema no pasa a la opción de dejar un mensaje. Si se quiere que esto ocurra debe ser implementado dentro del SCRIPT.

*NExt* es el número de extensión a donde pasar la llamada.

*Timeout* el número de segundos máximo a esperar.

*MensajeVoz* es el fichero de voz a pasar a la extensión. Si no se va a pasar ninguno debe valer \*.

*MensajeLCD* es una cadena que se mostrará en el display lcd de la extensión.

- Salidas anómalas:*
- 1.- Timeout en el último intento.
  - 2.- Comunicando en el último intento.
  - 4.- Extensión inexistente.

## **TRANSFER2EXTENSIONASK *N*Ext Timeout [*MensajeLCD*]**

*Descripción:* Transfiere la llamada a una extensión preguntando previamente el nombre del llamante por el canal 0. Al transferir la llamada ofrece al llamado la opción de responder pulsando 0, decir que no contesta pulsando 1 o decir que comunica pulsando cualquier otra tecla. Si consigue transferir la llamada finaliza la ejecución del script. Si la extensión no responde o comunica, el sistema no pasa a la opción de dejar un mensaje. Si se quiere que esto ocurra debe ser implementado dentro del SCRIPT.

*N*Ext es el número de extensión a donde pasar la llamada.

*Timeout* el número de segundos máximo a esperar.

*MensajeLCD* es una cadena que se mostrará en el display lcd de la extensión.

- Salidas anómalas:*
- 1.- Timeout en el último intento.
  - 2.- Comunicando en el último intento.
  - 4.- Extensión inexistente.

## **RESERVEEXTENSION *N*Ext [*N*Var *Etiqueta* [*Prioridad* [*Esperando*]]]**

*Descripción:* Comprueba si la extensión (o grupo) indicada está disponible. Si lo está salta inmediatamente a la instrucción indicada por *Etiqueta*. Si la extensión no está disponible, el script se coloca en una cola de espera para acceder a dicha extensión y en la variable indicada por *N*var se devuelve el número de posición en dicha cola. Se pueden incluir varios comandos RESERVEEXTENSION en un mismo script.

En el momento en que la extensión de un comando RESERVEEXTENSION se quede disponible para el script, se saltará a *Etiqueta* y se saldrá de la cola de espera de todos los comandos RESERVEEXTENSION que se hayan ejecutado. Hasta ese momento, la variable que se haya indicado está bloqueada para contener de forma dinámica el número de posición dentro de la cola de reserva. Hay que tener en cuenta que la extensión puede ser un grupo, en cuyo caso si se reservan de forma individual extensiones de dicho grupo, las colas de reserva serán independientes y no se garantiza cual de ellas tendrá prioridad.

Si el comando se ejecuta sólo con el número de extensión, lo que se hace es anular la reserva que se haya realizado anteriormente para esa extensión.

Se puede indicar una prioridad, esta prioridad es un número entre 1 (mayor prioridad) y 9 (menor prioridad), las reservas con mayor prioridad siempre se colocan en la cola delante de las de menor prioridad. Si no se indica prioridad se encolan con prioridad 1 (máxima).

*Esperando* nos indica si la reserva debe ser atendida o no, 0 indica que se atiende la reserva cuando le corresponda (este es el valor por omisión), y 1 indica

que no se atienda la reserva cuando le corresponda aunque la reserva sigue manteniendo su posición en la cola, para cambiar en una reserva el valor de este parámetro debemos volver a realizar la misma reserva con el nuevo valor de este parámetro.

Continuamente se va comprobando si las extensiones disponibles tienen alguna llamada en su cola, si es así se atiende dicha llamada. Después se va comprobando si los grupos tienen alguna extensión disponible y si tienen alguien en cola, si es así se van atendiendo las reservas por orden de cola, los grupos se van comprobando empezando por los de mayor prioridad, de esta forma si se queda disponible una extensión que pertenezca a varios grupos, primero atiende a las llamadas en cola de los grupos con mayor prioridad, aunque lleven menos tiempo en cola que otras llamadas en cola en grupos de menor prioridad.

En el caso de grupos, las extensiones no se consideran disponibles hasta que pase, como mínimo, desde que cuelguen o dejen de sonar sin descolgar el tiempo indicado, para este grupo, por el parámetro Delay\_Time, si no se ha indicado ningún valor para este grupo, se suponen disponibles inmediatamente después de colgar o dejar de sonar sin descolgar.

*Salidas anómalas:*        4.- Variable no existente (mayor que 39).  
                                  5.- Extensión inexistente.

## **CALCWAITTIME NVar NExt [Tmedio]**

*Descripción:* Devuelve en la variable local indicada el tiempo medio previsto de espera para ser atendido. Esta función sólo se puede llamar si se ha reservado previamente la extensión o grupo indicado con el comando RESERVEEXTENSION.

*NVar* es el número de variable donde devolver el resultado en segundos.

*NExt* es el número de grupo o extensión del que queremos hacer la predicción.

*Tmedio* es el tiempo medio de llamada con el que estamos trabajando. Si no se especifica se toma el valor por defecto que se haya asignado para el grupo en el fichero Sutil.ini. En caso de que sea una extensión suelta, ha de especificarse.

*Salidas anómalas:*        4.- Extensión o grupo no reservado previamente.

## **CREATETMPGROUP NumVar Distribución RetryCall ListaExtensiones**

*Descripción:* Nos permite crear temporalmente un grupo en un script, este grupo solo podrá ser utilizado en dicho script.

Una vez que en el script se hace a este grupo temporal un CALL2EXTENSION o un CALL2EXTENSIONBG y termina dicho comando, el grupo desaparece.

Este grupo temporal desaparece en el momento en que ocurre uno cualquiera de los siguientes eventos:



- Se ejecuta un CALL2EXTENSION o CALL2EXTENSIONBG con este número de grupo y el CALL2EXTENSION o CALL2EXTENSIONBG finaliza (con éxito o sin éxito).
- Se hace otro CREATETMPGROUP sin antes haber hecho un CALL2EXTENSION o CALL2EXTENSIONBG.
- El script finaliza.
- El script retorna a un script padre mediante la instrucción RETURN. Los grupos son locales al script. Esto quiere decir que si hacemos un CREATETMPGROUP en un script padre y llamamos a un hijo en éste podemos hacer otro CREATETMPGROUP. Al retornar al padre el grupo temporal todavía no habría sido destruido.

*NumVar* es el número de variable donde el comando nos devuelve el identificador de grupo que Sutil le asigna a este grupo temporal, este identificador de grupo tiene el formato GTxxx, donde xxx toma un valor entre 001 a 030.

En *Distribución* podemos indicar como queremos que se distribuya la llamada entre las extensiones que forman el grupo, para más información consultar el *Manual del Administrador de Sutil*, Tipo Distribución puede ser:

- 0 - Orden en el Grupo.
- 1 - Orden Correlativo.
- 2 - Todos a la vez.
- 3 - Más tiempo libre.

RetryCall puede valer:

- 0 - El Grupo se crea con la propiedad RetryCall no activa.
- 1 - El Grupo se crea con la propiedad RetryCall activa.

*ListaExtensiones* contiene la lista de extensiones que forman el grupo separadas por comas.

*Salidas anómalas:*

- 4.- Número de variable fuera de rango o tipo de distribución incorrecto.
- 5.- No se pueden crear más grupos temporales, en este momento están creados el máximo de grupos temporales.

## **2.8.- Comunicación a través de sockets IP.**

Con el fin de poder comunicarse con aplicaciones externas se provee de un sistema simple de intercambio de información a través de sockets. La aplicación externa deberá mantener el puerto abierto a la espera de recibir paquetes UDP con los comandos desde SUTIL.

Tipicamente el programa externo podrá realizar una búsqueda en una base de datos y/o algún tipo de cálculo con los parámetros. La aplicación externa recibirá peticiones a través de dicho socket que contestará de forma secuencial FIFO.

El programa externo deberá mandar la respuesta a la dirección IP y al puerto desde donde se ha recibido el paquete UDP, el puerto desde donde el comando SEND2SOCKET envía el paquete UDP varía en cada script que se ejecuta.

El comando OpenSocket abre un socket y dicho socket no se volverá a cerrar hasta la finalización de la ejecución del script.

Sólo se pueden utilizar números de socket del 1 al 10. El resto están reservados.

Los paquetes de datos enviados por un comando SEND2SOCKET, recibidos como respuesta a un comando SEND2SOCKET y los recibidos con destino a un comando ONSOCKET tiene el mismo formato salvo las diferencias que se indiquen de forma explícita:

Son del tipo UDP, siempre tienen 264 bytes de tamaño, y los 3 campos de los que constan y su formato son:

- IDENTIFICADOR: (entero de 4 bytes) Los paquetes recibidos por SUTIL con destino un comando ONSOCKET deben contener 99 en este campo.

Los paquetes recibidos por SUTIL como respuesta a un comando SEND2SOCKET deben contener 0 en este campo.

Los paquetes enviados por SUTIL debidos a un comando SEND2SOCKET contienen 1 en este campo.

- TAMAÑO: (entero de 4 bytes), este campo hoy día no se utiliza, por lo que su contenido es irrelevante.
- CONTENIDO: (cadena de 256 caracteres) contiene la cadena recibida o la que se quiere enviar terminada con un Byte cero (al estilo del lenguaje C).

En los paquetes recibidos, SUTIL introduce en la variable de respuesta ( del comando SEND2SOCKET u ONSOCKET, según proceda ) los caracteres del campo CONTENIDO hasta el primer Byte cero que encuentre.

En los paquetes enviados, Sutil incluye en este campo primero la cadena que se quiere enviar terminada con un Byte cero, y rellena hasta los 256 car. con caracteres sin especificar, esto quiere decir que es normal que nos encontremos restos de cadenas enviadas en anteriores SEND2SOCKET.

<b>NOTA:</b>	<p>Los campos IDENTIFICADOR y TAMAÑO, son enteros de 4 Bytes en formato <i>little-endian</i>, es decir que el primer Byte es el LSB (Least Significant Byte, Byte menos significativo).</p> <p>Es importante aclarar que hablamos del orden de los Bytes, no del orden de los bit en cada Byte, en cada Byte siempre el primer bit es el más significativo.</p> <p>Podemos verlo más claro si lo comparamos con la diferencia que hay al escribir en árabe o castellano. Los árabes lo hacen de derecha a izquierda al contrario que nosotros, pero la palabra las escriben como nosotros de izquierda a derecha.</p>
--------------	---

**Ejemplo:**

En el siguiente paquete de ejemplo se envía la palabra: "REQUEST"

**Paquete completo:**

```
01 00 00 00 07 00 00 00 52 45 51 55 45 53 54 00 34 2E 31 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

**Paquete desglosado:**

01 00 00 00 --> Identificador, 4 bytes. Sutil siempre envía un 1 con un SEND2SOCKET desde el script.

07 00 00 00 --> Tamaño del campo *Contenido*, 4 bytes. Si el hexadecimal recibido fuera AB CD EF GH, leyendolo al revés quedaría GH EF CD AB. En el ejemplo esto se traduce en 00 00 00 07, que en decimal coincide con el número 7 (longitud de la palabra REQUEST que es la que se envía)

```
52 45 51 55 45 53 54 00 34 2E 31 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

00 00 00 --> Contenido del paquete: 52 en hexadecimal es R, 45 es E, 51 es Q, ... hasta que encuentra un carácter 00 , el resto se desecha.

## **OPENSOCKET *Handle DireccionIP Puerto***

*Descripción:* Abre un socket para poder comunicarse con un programa externo.

DireccionIP y Puerto son los del destino donde esta escuchando el programa externo, y donde los comandos SEND2SOCKET enviarán los paquetes UDP, no se deben confundir con la dirección IP y el puerto desde donde el comando SEND2SOCKET envía los paquetes UDP, evidentemente la dirección IP desde donde el comando SEND2SOCKET envía los paquetes es la dirección IP de Sutil y el puerto lo determina el S.O. la primera vez que se ejecuta el comando SEND2SOCKET en cada script que se ejecuta, el S.O. va dando puertos consecutivos desde el 1025 que estén libres.

*Salidas anómalas:*        5.- No se puede abrir el socket.  
                                  7.- Handle en uso.

## **SEND2SOCKET *Handle NVarRespuesta Timeout CadenaEnvio***

*Descripción:* Envía un paquete UDP (según el formato indicado anteriormente) conteniendo *CadenaEnvio* a la dirección Ip y al puerto establecido por el comando OPENSOCKET correspondiente al *Handle* indicado.

Después se queda a la escucha (el tiempo indicado por Timeout) en el puerto por el que se ha enviado el paquete UDP y esperando una respuesta que almacenará en la variable indicada.

Una vez transcurrido el tiempo indicado en timeout sin recibir una respuesta, el comando deja de escuchar en dicho puerto, y termina devolviendo el error 1. En este caso se genera una traza en el fichero Trazas.log, si no queremos que se genere dicha traza indicaremos el timeout terminando en N, es decir 10 significa timeout de 10 seg. Y 10N significa también un timeout de 10 seg. pero que no dejara traza en caso de haber timeout.

Es importante recalcar que el puerto donde se queda a la escucha el comando SEND2SOCKET es el puerto desde donde se ha enviado el paquete UDP, que no tiene nada que ver con el puerto destino a donde se ha enviado el paquete UDP. El puerto desde donde se envía el paquete UDP lo determina el S.O. la primera vez que se ejecuta el comando SEND2SOCKET en cada script que se ejecuta, el S.O. va dando puertos consecutivos desde el 1025 que estén libres.

*Handle* es el asignado en el comando OPENSOCKET.

*NVarRespuesta* vale entre 0 y 39 e indica donde almacenar el resultado.

*Timeout* indica el número de segundos para esperar la respuesta.

*CadenaEnvio* es la cadena a enviar y se toma hasta el final de la línea o de un carácter ";" (comienzo de comentario) si lo contiene dicha línea, incluyendo los espacios o tabuladores que pueda contener.

*Salidas anómalas:*

- 1.- Timeout.
- 4.- Variable no existente (mayor que 39).  
No se ha hecho un OPENSOCKET de este Handle.
- 5.- Error en el envío/recepción (error de red, ...).

### **ONSOCKET Handle [ NvarRespuesta Etiqueta ]**

*Descripción:* En el caso de que se reciba algo por el socket especificado por Handle, siempre y cuando no sea como respuesta a una petición previa efectuada con SEND2SOCKET, se almacena lo recibido en la variable indicada y se salta a la etiqueta indicada. Si no se indica etiqueta, ni NvarRespuesta, se desactiva dicho salto.

*Salidas anómalas:*

- 4.- Variable no existente (mayor que 39).
- 5.- Socket no abierto.

## **2.9.- Gestión de interrupciones.**

Existen tres instrucciones que alteran la ejecución secuencial de un script: ONHOOK, ONSOCKET y RESERVEEXTENSION. Estas instrucciones generan lo que denominamos una INTERRUPCIÓN.

Antes de la ejecución de cada instrucción se comprueba si se ha producido alguna interrupción. Si así ha sido, en lugar de ejecutar la instrucción que correspondía, se desvía la ejecución hasta la etiqueta especificada por la instrucción correspondiente. Esta instrucción se ejecutará sin volver a comprobar interrupciones hasta la siguiente instrucción.

Utilizando la instrucción DISABLEINT deshabilitamos temporalmente la comprobación de las interrupciones, aunque estas quedan almacenadas en una cola que será ejecutada al volver a activar las interrupciones con el comando ENABLEINT.

Si en el momento de comprobar las interrupciones se han producido varias de estas, sólo se ejecutará una de ellas, según un orden de prioridad establecido. Esto significa que se ejecutará al menos la primera instrucción de la rutina de interrupción establecida. En la ejecución de la siguiente instrucción se volverán a comprobar las interrupciones y si la instrucción que se ha llegado a ejecutar no ha sido DISABLEINT, se saltará a la etiqueta establecida por la siguiente interrupción en orden de prioridad. Es por ello, que el uso natural de la instrucción DISABLEINT es como primera instrucción de rutinas de proceso de interrupciones.

El orden de prioridad de evaluación de las interrupciones es el siguiente:

- ONHOOK
- RESERVEEXTENSION
- ONSOCKET

En general, no existe un orden garantizado para la evaluación de interrupciones del mismo nivel. Es decir, si ocurren varios cuelgues de canales con interrupción programada (ONHOOK), se ejecutará uno de ellos, y en la siguiente instrucción o cuando vuelvan a estar disponibles las interrupciones si ésta es un DISABLEINT se ejecutará la siguiente interrupción.

Solamente se garantiza una prioridad en la comprobación de extensiones con RESERVEEXTENSION. El orden de verificación de estas interrupciones es el siguiente: primero se verifican las extensiones sueltas y después los grupos según el orden de prioridad que se haya establecido. Es decir, si al mismo tiempo se quedan libres extensiones de dos grupos diferentes, se ejecutará el RESERVEEXTENSION asociado a la cola de mayor prioridad.

En el momento que se produce una interrupción RESERVEEXTENSION, se eliminan todo el resto de comandos RESERVEEXTENSION. Eso quiere decir, que ya no se encolarán más interrupciones de este tipo. Solamente se llegará a ejecutar la primera que haya sido evaluada. Bien, en el momento de producirse, bien cuando las interrupciones sean habilitadas, si se encontraban deshabilitadas en el momento de la interrupción.

Durante el tiempo en que las interrupciones están deshabilitadas a través del comando `DISABLEINT`, las interrupciones que ocurran son encoladas para su posterior ejecución al volver a habilitar las interrupciones. Cuando éstas se habilitan con el comando `ENABLEINT`, el salto se producirá a la etiqueta que esté establecida en ese momento, no a la que estuviera establecida en el momento que ocurrió la interrupción. Eso quiere decir que ejecutando un comando sin especificar etiqueta estaremos "borrando" dichas interrupciones, ya que no producirán ningún salto al restaurarse las interrupciones.

Hay varias instrucciones que pueden ser ejecutadas a lo largo de varios ciclos lógicos de SUTIL. Entre estas instrucciones, en varias de ellas se comprueba la existencia de interrupciones en cada uno de sus ciclos, por lo que dichas instrucciones pueden ser interrumpidas durante el transcurso de su ejecución sin que llegue a finalizar. Si se quiere garantizar su completa ejecución, se deberá colocar una instrucción `DISABLEINT` de forma previa. Dichas instrucciones son las siguientes:

- `CALL2EXTENSION`
- `CALL2EXTERIOR`
- `INPUT`, `INPUTR`
- `PLAYFILE`, `PLAYM`, `PLAYTIME`, `PLAYNUMBER`
- `RECORDFILE`
- `BEEP`
- `PAUSE`

Si la interrupción que se produce es un cuelgue de un canal, y el comando que se está ejecutando está utilizando dicho canal, si está deshabilitado el `ONHOOK`, el comando dará un error 4.- canal no abierto (además de generar una traza de nivel 3), por lo que se recomienda tener activo el `ONHOOK` adecuado para el caso de que se produzca el cuelgue del canal durante la ejecución de este comando.

Durante la ejecución de un script llamado con la instrucción `CALL`, se ejecutarán las interrupciones locales que se definan en dicho proceso. Una vez que se retorne al proceso padre se verificarán las interrupciones definidas en el proceso padre y se ejecutarán. Por ejemplo si ocurre el cuelgue de un canal durante la ejecución de un script llamado con `CALL`, se ejecutará el `ONHOOK` local asociado al proceso hijo primero y cuando se finalice dicho script y se retorne, se ejecutará el del proceso padre.

## **DISABLEINT**

*Descripción:* Desactiva temporalmente la ejecución de interrupciones, quedando estas encoladas para su posterior ejecución. Garantizando que no se altere la ejecución secuencial de instrucciones, salvo por los comandos de salto ordinarios.

## **ENABLEINT**

*Descripción:* Reactiva las interrupciones después de la ejecución previa de un comando DISABLEINT. Si ha ocurrido alguna interrupción durante el período de tiempo en que estas han estado deshabilitadas, en el siguiente ciclo de interrupción se ejecutará la de mayor prioridad.



## **2.10.- Gestión de buzones vocales.**

SUTIL dispone de un sistema de correo vocal interno que permite la existencia de hasta un millón de buzones vocales simultáneos dentro de cada grupo. Puede haber varios grupos de buzones separados en un mismo equipo SUTIL. En configuración se especifica el número de grupos que existirán y sus nombres. Por defecto, a cada extensión dinámica se le asigna un buzón dentro del grupo 0 cuyo identificador es el mismo que el de la propia extensión. Por ello, los buzones del 1 al 9.999 del grupo 0 se encuentran reservados para su asociación a las extensiones.

Para comenzar a utilizar el sistema de buzones en un script hay que empezar ejecutando el comando MBOXOPEN, en el que se especifica el grupo de buzones a utilizar.

Los buzones de voz contienen mensajes no-leídos y archivados. Cuando un mensaje es escuchado de forma completa por el usuario pasa a estar archivado de forma automática.

### **MBOXOPEN NGrupo**

*Descripción:* Pone en marcha el sistema de buzones para que pueda ser utilizado dentro del script. Se le especifica el grupo de buzones que se desea utilizar.

*Salidas anómalas:* 1.- Módulo Mbox no disponible.  
4.- Grupo inexistente.

### **MBOXCLOSE**

*Descripción:* Acaba la utilización del sistema de buzones dentro del script en que se ejecute. A partir de ese momento ya no se puede utilizar dentro del script hasta que no se haga de nuevo un MBOXOPEN.

*Salidas anómalas:* 1.- Módulo Mbox no disponible.

### **MBOXCREATE NBuzon NumVar Clave Email**

*Descripción:* Crea un buzón nuevo. Se le asignará el número de buzón indicado. Si se indica 0 como número de buzón, se le asignará un nuevo número de buzón. En el caso de que el grupo abierto sea el 0, el número asignado estará por encima de 10.000. En la variable indicada se devuelve el número de buzón finalmente asignado. Tendrá un mensaje de saludo estandar. Para modificarlo hay que utilizar el comando MBOXRECORDWM. Se puede especificar una dirección de e-mail asociada al buzón.

*Salidas anómalas:* 4.- Buzón ya existente.

**MBOXMENU Canal NBuzon Clave**

*Descripción:* Reproduce los mensajes grabados en el buzón, accediendo al menú de reproducción de los mismos. Si Clave toma el valor X significa que el comando debe pedir él mismo la clave al usuario y si vale L significa que debe pedirla si el buzón está configurado para pedirla en modo local y no debe pedirla ni comprobarla si está configurado para no pedirla en modo local.

*Salidas anómalas:*

- 4.- Buzón inexistente o canal no abierto.
- 5.- Recurso de voz no disponible.
- 6.- Clave errónea.

**MBOXNUMARC NBuzon NVar**

*Descripción:* Retorna el número de mensajes archivados del buzón indicado en la variable indicada.

*Salidas anómalas:*

- 4.- Buzón inexistente.

**MBOXNUMNEW NBuzon NVar**

*Descripción:* Retorna el número de mensajes no-leídos del buzón indicado en la variable indicada.

*Salidas anómalas:*

- 4.- Buzón inexistente.

**MBOXRECORDM Canal NBuzon Opcion1 Opcion2**

*Descripción:* Graba un mensaje en un buzón.

*NBuzon* es el número de buzón.

*Opcion1* puede valer 0 o 1. 1 significa que se reproduzca el mensaje de invitación a dejar mensaje en el buzón, 0 que no se haga esto.

*Opcion2* puede valer 0 o 1. 1 significa que se escuche por los altavoces del sistema lo que se esté grabando, 0 que no se haga esto.

*Salidas anómalas:*

- 4.- Buzón inexistente o canal no abierto.
- 5.- Recurso de voz no disponible.

**MBOXRECORDWM Canal NBuzon Clave Opcion**

*Descripción:* Graba el mensaje de bienvenida de un buzón.

*Canal* es el número de canal desde donde se graba el mensaje.

*NBuzon* es el número de buzón.

*Clave* es la clave del buzón. Si toma el valor X significa que el comando debe pedir él mismo la clave al usuario y si vale L significa que debe pedirla si el buzón

está configurado para pedirla en modo local y no debe pedirla ni comprobarla si está configurado para no pedirla en modo local.

*Opcion* puede valer 0 o 1. 0 es la opción por defecto. 1 significa que se reproduzca el mensaje de invitación a dejar el mensaje de bienvenida. 0 que no se haga esto.

*Salidas anómalas:* 4.- Buzón inexistente o canal no abierto.  
5.- Recurso de voz no disponible.  
6.- Clave errónea.

### **MBOXPLAYWM Canal NBuzon**

*Descripción:* Reproduce el mensaje de bienvenida a un buzón de voz.

*Salidas anómalas:* 4.- Buzón inexistente o canal no abierto.  
5.- Recurso de voz no disponible.

### **MBOXREMOVE Nbuzon Clave**

*Descripción:* Elimina un buzón.

*Salidas anómalas:* 4.- Buzón no existente.  
6.- Clave errónea.

### **MBOXMAX NumVar**

*Descripción:* Retorna en la variable indicada el número máximo de buzón asignado hasta el momento.

### **MBOXEXIST NBuzon NVar**

*Descripción:* Retorna en la variable indicada 1 si el buzón indicado existe y 0 si no existe.

### **MBOXEMAIL NBuzon NVar**

*Descripción:* Retorna el contenido del campo Email del buzón indicado en la variable indicada.

*Salidas anómalas:* 4.- Buzón no existente.

## 2.11.- Contadores externos.

Los contadores proporcionan una forma de obtener números únicos de referencia auto-incrementales. Son útiles, por ejemplo, para utilizarlos como parte del nombre de ficheros que se vayan grabando con información recogida a través de un sistema IVR.

Se guardan en disco, por lo que aunque el motor de SUTIL se reinicie mantienen su valor. Se garantiza el bloqueo en el acceso, por lo que con el uso del comando COUNTER\_INC nunca se obtendrán dos valores iguales aunque se llamen simultáneamente desde dos scripts distintos.

Se pueden crear contadores desde el 1 hasta el 999.999.999 (inclusive).

Los contadores pueden tomar valores entre 0 y 999.999.999.999.999.999 (inclusive).

### COUNTER\_INIT

*Descripción:* Debe estar al comienzo de todo script que utilice los contadores.

*Salidas anómalas:*

- 5.- Error al inicializar Módulo Contadores.
- 7.- Módulo Contadores ya inicializado anteriormente.

### COUNTER\_RESET Contador NumVar [ValorInicio]

*Descripción:* Inicializa el contador especificado con el valor que se indica, si no existe el contador indicado lo crea.

Si todo ha ido correctamente nos devuelve 0 en la variable *Numvar*.

*Contador* debe estar comprendido entre 1 y 999.999.999, para valores fuera de este rango se devuelve ERR en la variable *NumVar*.

*ValorInicio* debe estar comprendido entre 0 y 999.999.999.999.999.999, para valores fuera de este rango se devuelve ERR en la variable *NumVar*.

*Salidas anómalas:*

- 1.- Módulo Contadores no disponible.
- 4.- Variable no existente (mayor que 39).
- 5.- Módulo Contadores no abierto.

*Ejemplos:*

– Exista o no el contador 1, y tenga el valor que tenga:

```
COUNTER_RESET 1 9 0
```

En la variable %v9 nos devuelve 0 y el contador 1 toma el valor 0.

```
COUNTER_RESET 1 9 28
```

En la variable %v9 nos devuelve 0 y el contador 1 toma el valor 28.

```
COUNTER_RESET 1 9
```

En la variable %v9 nos devuelve 0 y el contador 1 toma el valor 0.

```
COUNTER_RESET 1 9 999999999999999999
```

En la variable %v9 nos devuelve ERR.

– En cualquier caso:

```
COUNTER_RESET 999999999 8
```

En la variable %v8 nos devuelve ERR.

```
COUNTER_RESET 999999999 8 2
```

En la variable %v8 nos devuelve ERR.

## **COUNTER\_INC Contador NumVar [NumDigitos]**

*Descripción:* Incrementa el valor actual del contador y lo devuelve en la variable *NumVar*, si no existe el contador indicado lo crea y devuelve 1.

Este es el comando que debe utilizarse normalmente para la asignación de número únicos.

*NumDigitos* indica con cuantos dígitos (completado con ceros a la izquierda), como mínimo, nos devuelve el valor del contador.

*Contador* debe estar comprendido entre 1 y 999.999.999, para valores fuera de este rango se devuelve ERR en la variable *NumVar*.

*Salidas anómalas:* 1.- Módulo Contadores no disponible.

4.- Variable no existente (mayor que 39).

5.- Módulo Contadores no abierto.

*Ejemplos:*

– Si el contador 1 tiene de valor actual 4:

```
COUNTER_INC 1 9
```

En la variable %v9 nos devuelve 5 y el contador 1 toma el valor 5.

```
COUNTER_INC 1 9 2
```

En la variable %v9 nos devuelve 05 y el contador 1 toma el valor 5.

– Si el contador 3 tiene de valor actual 237:

```
COUNTER_INC 3 9
```

En la variable %v9 nos devuelve 238 y el contador 3 toma el valor 238.

```
COUNTER_INC 3 9 2
```

En la variable %v9 nos devuelve 238 y el contador 3 toma el valor 238.

COUNTER\_INC 3 9 4

En la variable %v9 nos devuelve 0238 y el contador 3 toma el valor 238.

- Si no existe el contador 5:

COUNTER\_INC 5 9

En la variable %v9 nos devuelve 1 y el contador 5 toma el valor 238.

## **COUNTER\_READ Contador NumVar [NumDigitos]**

*Descripción:* Lee el valor actual del contador en la variable indicada. Este valor se obtiene sólo a modo informativo. Debe utilizarse el comando COUNTER\_INC para garantizar su unicidad.

*NumDigitos* indica con cuantos dígitos (completado con ceros a la izquierda), como mínimo, nos devuelve el valor del contador.

*Contador* debe estar comprendido entre 1 y 999.999.999, para valores fuera de este rango se devuelve ERR en la variable *NumVar*.

*Salidas anómalas:*

- 1.- Módulo Contadores no disponible.
- 4.- Variable no existente (mayor que 39).
- 5.- Módulo Contadores no abierto.

*Ejemplos:*

- Si el contador 1 tiene de valor actual 4:

COUNTER\_READ 1 9

En la variable %v9 nos devuelve 4 y el contador 1 continúa con el valor 4.

COUNTER\_READ 1 9 2

En la variable %v9 nos devuelve 04 y el contador 1 continúa con el valor 4.

- Si el contador 1 tiene de valor actual 237:

COUNTER\_READ 1 9

En la variable %v9 nos devuelve 237 y el contador 1 continúa con el valor 237.

COUNTER\_READ 1 9 2

En la variable %v9 nos devuelve 237 y el contador 1 continúa con el valor 237.

COUNTER\_READ 1 9 4

En la variable %v9 nos devuelve 0237 y el contador 1 continúa con el valor 237.

- Si el contador 5 no existe:

COUNTER\_READ 5 9

En la variable %v9 nos devuelve ERR.

## **2.12.- Generación del histórico de llamadas.**

El histórico de llamadas de SUTIL genera una o varias líneas asociadas a cada llamada entrante, saliente o interna que se produce. Cada una de estas líneas es lo que se denomina un segmento de llamada. Una llamada interna siempre está compuesta por un único segmento. Pero las llamadas internas o externas pueden estar formadas por más de un segmento.

En dicho histórico existe un campo reservado para su asignación a través del comando SETCODECALL. Sus usos son múltiples. Por ejemplo, puede servir para dejar registrado el tipo de servicio que se ha proporcionado al cliente o en el caso de sistemas de tarjetas pre-pago para dejar registrado el número de tarjeta utilizado en la transacción.

En los casos en que SETCODECALL se utilice para registrar el tipo de servicio proporcionado, es posible que una vez finalizado el proceso IVR por el que se proporciona dicho servicio se ofrezca la posibilidad de volver al principio. Con el fin de indicar la prestación de un nuevo servicio diferente puede ser conveniente fraccionar la llamada en varios segmentos diferentes en el histórico. Esto se puede hacer a través del comando GENCALLREG.

### **SETTYPECALL Canal Tipo**

*Descripción:* Asigna un tipo de llamada, según se haya llegado a atender la llamada o no, a la llamada entrante asociada al canal especificado. El tipo de llamada se graba en el segmento actual y todos los siguientes hasta el fin de la llamada o hasta que vuelva a ser modificado con otro comando SETTYPECALL.

*Tipo* puede tomar uno de estos valores:

0.- Sin especificar.

1.- Llamada atendida. Normalmente lo utilizamos para indicar que la llamada entrante se ha llegado a conectar con una extensión.

2.- Llamada abandonada en cola. Normalmente lo utilizamos para indicar que el llamante colgó antes de que llegáramos a conectar la llamada entrante con una extensión. Tanto en el caso de que hayamos llegado a descolgar como si no hemos llegado a descolgar.

3.- Llamada rechazada por saturación. Normalmente lo utilizamos para indicar que hemos rechazado la llamada porque vemos que no podemos atenderla

*Salidas anómalas:* 4.- Canal no abierto.

### **SETCODECALL Canal Código**

*Descripción:* Asigna un código que se grabará en el registro del histórico de la llamada asociada con *Canal*, con el fin de que más tarde se puedan obtener listados y estadísticas relacionados con él. Este código se graba en el segmento actual y todos los siguientes hasta el final de la llamada o hasta que vuelva a ser modificado con otro comando SETCODECALL.



En el fichero se almacena con un tamaño de 20 car. si el tamaño de la cadena que le asignamos en *Codigo* es mayor, se truncara al almacenarse.

*Salidas anómalas:* 4.- Canal no abierto.

## **SETLANGCALL Canal CodIdioma**

*Descripción:* Asigna el código del idioma utilizado en la llamada, que será grabado en el campo correspondiente del fichero histórico. Este código de idioma se graba en el segmento actual y en los siguientes hasta el final de la llamada o hasta que se vuelva a modificar con el comando SETLANGCALL. El fin de grabar este código es el de poder realizar estadísticas de los idiomas utilizados en las llamadas.

En el *Manual del Administrador de Sutil*, en el capítulo de Datos Técnicos podemos consultar la tabla de Idiomas.

*Salidas anómalas:* 4.- Canal no abierto.

## **GENCALLREG Canal**

*Descripción:* Hace que en la llamada asociada con *Canal* se cierre el segmento actual con la duración realizada hasta ese momento y con los campos de tipo, código e idioma según se hayan asignado con los comando SETTYPECALL, SETCODECALL y SETLANGCALL. A partir de ese momento se generará un nuevo segmento de llamada.

*Salidas anómalas:* 4.- Canal no abierto.

## **GETNUMCALL Canal NvarLla NvarSeg NvarGra**

*Descripción:* Nos da información sobre la llamada que se esta cursando en ese momento en *Canal*.

Se puede utilizar sobre cualquier canal abierto, y también se puede utilizar sobre el canal 0 antes de que la llamada sea descolgada con INCOMINGCALL.

*NvarLla* es el número de variable donde nos devuelve el número de llamada que se le ha asignado, en el fichero histórico de llamadas.

*NvarSeg* es el número de variable donde nos devuelve el número de segmento de llamada que se le ha asignado, en el fichero histórico de llamadas.

*NvarGra* es el número de variable donde nos devuelve el número que se le ha asignado al fichero donde se esta grabando dicha llamada, en caso de que no se este grabando devuelve 0.

*Salidas anómalas:* 4.- Canal no abierto o nº de variable no válido .

## 2.13.- Lista negra de teléfonos.

Disponemos de dos comandos para poder añadir y borrar números de teléfonos de una lista y otro para poder consultar si un número de teléfono esta incluido en dicha lista.

Esto es util para poder confeccionar una lista, por ejemplo, de telefonos de los que no queremos atender las llamadas que vengan de ellos.

Existe también en el administrador de Sutil un punto de mantenimiento de esta lista.

### BL\_INIT

*Descripción:* Debe estar al comienzo de todo script que utilice la lista negra.

*Salidas anómalas:* 5.- Error al inicializar Módulo Lista Negra.  
7.- Módulo Lista Negra ya inicializado anteriormente.

### BL\_INSERT NumTelefono NumVar

*Descripción:* Inserta el número de teléfono indicado en *NumTelefono* en la lista negra.

Si todo ha ido correctamente nos devuelve 0 en la variable *Numvar*.

Si ya existe ese número de teléfono en la lista nos devuelve ERR en la variable *Numvar*.

Si se produce cualquier error al consultar la base de datos nos devuelve ERR en la variable *Numvar*.

*Salidas anómalas:* 1.- Módulo Lista Negra no disponible.  
4.- Variable no existente (mayor que 39).  
5.- Módulo Lista Negra no abierto.

*Ejemplos:*

– No existe el número 952653516 en la lista:

```
BL_INSERT 952653516 9
```

En la variable %v9 nos devuelve 0 y se añade el número 952653516 a la lista.

– Si existe el número 952653516 en la lista:

```
BL_INSERT 952653516 9
```

En la variable %v9 nos devuelve ERR y el número 952653516 permanece en la lista.

**BL\_EXIST NumTelefono NumVar**

*Descripción:* Comprueba si el número de teléfono indicado en *NumTelefono* esta contenido en la lista negra.

Si no existe el número de teléfono en la lista negra nos devuelve 0 en la variable *Numvar*.

Si existe el número de teléfono en la lista negra nos devuelve 1 en la variable *Numvar*.

*Salidas anómalas:*

- 1.- Módulo Lista Negra no disponible.
- 4.- Variable no existente (mayor que 39).
- 5.- Módulo Lista Negra no abierto.

*Ejemplos:*

- No existe el número 952653516 en la lista:

```
BL_EXIST 952653516 9
En la variable %v9 nos devuelve 0.
```

- Si existe el número 952653516 en la lista:

```
BL_EXIST 952653516 9
En la variable %v9 nos devuelve 1.
```

**BL\_REMOVE NumTelefono NumVar**

*Descripción:* Borra el número de teléfono indicado en *NumTelefono* de la lista negra.

Si todo ha ido correctamente nos devuelve 0 en la variable *Numvar*.

Si no existe ese número de teléfono en la lista nos devuelve ERR en la variable *Numvar*.

Si se produce cualquier error al consultar la base de datos nos devuelve ERR en la variable *Numvar*.

*Salidas anómalas:*

- 1.- Módulo Lista Negra no disponible.
- 4.- Variable no existente (mayor que 39).
- 5.- Módulo Lista Negra no abierto.

*Ejemplos:*

- No existe el número 952653516 en la lista:

```
BL_REMOVE 952653516 9
En la variable %v9 nos devuelve ERR.
```

- Si existe el número 952653516 en la lista:

```
BL_REMOVE 952653516 9
```

En la variable %v9 nos devuelve 0 y borra el número 952653516 de la lista.

## 2.14.- Generación de fichero de trazas.

### OPENFILE *Handle Modo Fichero*

*Descripción:* Abre un fichero para escribir trazas en él.

*Handle* lo utiliza el comando WRITE2FILE para indicar en que fichero escribe la traza.

Los valores válidos para *Handle* son entre 1 y 100.

*Modo* puede valer 0 ó 1 :

0 Al comienzo de cada traza no se añade la fecha y hora actual.

1 Al comienzo de cada traza se añade la fecha y hora actual.

*Fichero* se genera por defecto en *Sutil\* , y puede incluir un camino hasta el fichero, en cualquier caso se parte del directorio *Sutil\* , salvo que *Fichero* comience por:

“\” en este caso se parte del directorio raíz de la unidad donde se este ejecutando Sutil.exe.

“letra de unidad:” en este caso, se indique explícitamente o no, siempre se parte del directorio raíz de la unidad indicada.

“\\Nombre\_ordenador\Recurso\_compartido” o  
“\\Dir\_IP\Recurso\_compartido” en este caso se parte del directorio raíz del Recurso\_compartido.

En todos estos casos podemos utilizar “\” o “/” indistintamente.

*Salidas anómalas:* 5.- Handle no válido (fuera del rango 1-100)  
7.- Handle ya en uso.

Ejemplos:

```
OPENFILE 1 0 Trazas.txt
```

Se genera el fichero: C:\Sutil\Trazas.txt

```
OPENFILE 1 0 C:\Trazas.txt
```

Se genera el fichero: C:\Trazas.txt

```
OPENFILE 1 0 Config\Trazas.txt
```

Se genera el fichero: C:\Sutil\Config\Trazas.txt

```
OPENFILE 1 0 ..\eisw\Trazas.txt
```

Se genera el fichero: C:\eisw\Trazas.txt

### **WRITE2FILE *Handle Cadena***

*Descripción:* Escribe una traza en un fichero.

*Handle* indica en que fichero escribe la traza, este fichero se debe abrir previamente con el comando OPENFILE.

*Cadena* es la traza a escribir en el fichero.

*Salidas anómalas:*

- 4.- No se ha hecho un OPENFILE de este Handle.
- 5.- Error al escribir (Disco duro lleno, error al acceder a unidad de red, ...).

### **WRITE2TRACE *Handle NivelError Cadena***

*Descripción:* Escribe una traza en un fichero con el formato para que la aplicación ei-Baliza Agente la mande al servidor ei-Central.

*Handle* indica en que fichero escribe la traza, este fichero se debe abrir previamente con el comando OPENFILE, indicando en el comando OPENFILE el modo 2.

*NivelError* es el nivel con el que se genera la traza para su posterior tratamiento por parte del ei-Baliza Agente.

*Cadena* es la traza a escribir en el fichero.

La traza que se genera tiene el formato:

```
AAAA/MM/DD HH:MM:SS EIM-108.09.21015.NivelError  
Cadena
```

*Salidas anómalas:*

- 4.- No se ha hecho un OPENFILE de este Handle.
- 5.- Error al escribir (Disco duro lleno, error al acceder a unidad de red, ...).

### ***3.- Funciones de Sutil asociadas a un script.***

Algunas de las operaciones ejecutables desde un terminal telefónico conectado a una extensión y algunas de las que se pueden ejecutar desde el monitor de estado de Sutil, lo que hacen es ejecutar un scripts, esto quiere decir que podemos modificar el comportamiento de dichas operaciones modificando el script.

Todos estos scripts estan en el directorio SysScr dentro del directorio donde este instalado Sutil ( Normalmente C:\Sutil ).

### **3.1.- Operaciones ejecutables desde un terminal telefónico conectado a una extensión que ejecutan un script.**

#### **3.1.1.- Asignar número de extensión lógica. (LOGIN).**

Código por defecto: \*28 + número de extensión

El script que se ejecuta es: AutenticacionInterna.ivr

#### **3.1.2.- Desasignar número de extensión lógica. (LOGOUT).**

Código por defecto: \*29 + número de extensión

El script que se ejecuta es: DesautenticacionInterna.ivr

#### **3.1.3.- Escuchar los mensajes grabados en nuestro buzón.**

#

El script que se ejecuta es: MbMenuInterno.ivr

#### **3.1.4.- Escuchar los mensajes grabados en otro buzón.**

Código por defecto: \*41 + número de buzón + #

El script que se ejecuta es: MbMenuInterno.ivr

#### **3.1.5.- Grabar mensaje de bienvenida de nuestro buzón.**

Código por defecto: \*42

El script que se ejecuta es: MbRecordWMInterno.ivr

#### **3.1.6.- Dejar un mensaje a un buzón concreto.**

Código por defecto: \*48 + número de buzón + #

El script que se ejecuta es: MbRecordMInterno.ivr

#### **3.1.7.- Pasar llamada a una extensión concreta sin esperar.**

Código por defecto: \*12 + número de extensión

El script que se ejecuta es: TransfDirecta.ivr

Puede indicarse un número de extensión o un número de grupo.

En el caso de que se indique un número de grupo, Sutil resuelve a que extensión llamar y al script se le pasa como parámetro la extensión en la que se ha resuelto y no el grupo.



### **3.1.8.- Ejecutar script de usuario.**

Código por defecto: \*14

El script que se ejecuta es: UserDefined.ivr

En el canal 0 es la extensión que ha ejecutado el script.

Si la extensión, en el momento de ejecutar este script, tiene alguien retenido, el canal retenido es el canal 1 en el script.

## **3.2.- Funciones ejecutables desde el Monitor de Estado de Sutil que ejecutan un script.**

### **3.2.1.- Retener llamada.**

El script que se ejecuta es: Retencion.ivr

Para retener una llamada, una vez que estamos hablando con ella, lo haremos pulsando otra vez la casilla de Hablar con Linea, que es la primera casilla que aparece en su fila. Al realizar esta acción se ejecuta el script Retencion.ivr situado en la carpeta Sutil\SysScr, pasandole el canal de entrada, dándonos la posibilidad de realizar acciones sobre él.

### **3.2.2.- Transferir llamada.**

El script que se ejecuta es: TransfMasDirecta.ivr

#### **4.- Ficheros Scripts de ejemplo.**

A continuación incluimos algunos scripts de ejemplo, agrupados por su campo de aplicación.

Las extensiones y grupos que se utilizan en estos ejemplos son:

Extensiones 201 a 204 están asignadas a las extensiones físicas 1 a 4 respectivamente.

Extensiones 205 a 208 no están asignadas a extensiones físicas.

Grupo 200 incluye las extensiones 201 a 204.

## **4.1.- Como centralita de empresa.**

### **4.1.1.- Todos nuestros operadores están ocupados. Espere unos momentos.**

Este script atiende las llamadas y las pone en una cola de espera, se van atendiendo por el orden en el que han llegado.

Mientras esperan a ser atendidos se les reproduce ciclicamente un mensaje y tono de marcar.

Para probar este script se puede llamar al 951 088 535

```
; OpVirtual_1.ivr
```

```
; Se descuelga la llamada.
```

```
INCOMINGCALL 0
```

```
; Reproduce por el canal 0 el fichero Bienvenida.wav
```

```
PLAYFILE 0 OpVirtual\Bienvenida 1
```

```
:INICIO
```

```
; Se reserva la extensión o grupo 200, en la variable %v0 nos va devolviendo el puesto en la cola, cuando se quede libre una extensión del grupo 200 salta a la etiqueta TRANSFERIR.
```

```
RESERVEEXTENSION 200 0 TRANSFERIR
```

```
:CICLO
```

```
; Reproduce por el canal 0 el fichero Espera.wav y un tono de marcar durante 10 segundos ciclicamente.
```

```
; Espera.wav "Espere por favor, en estos momentos todos nuestros operadores estan ocupados. Su llamada sera atendida lo antes posible".
```

```
PLAYFILE 0 OpVirtual\Espera 1
```

```
PLAYTONE 0 0
```

```
PAUSE 10
```

```
GOTO CICLO
```

```
:TRANSFERIR
```

```
CALL2EXTENSION 1 200 10
```

```
IFERR 0
```

```
CONNECT 0 1
```

```
EXIT
```

```
ENDIFERR
```

```
; Si no tiene éxito la llamada al grupo vuelve al inicio.
```

```
GOTO INICIO
```

#### **4.1.2.- Se pide una opción y si no se pulsa nada se pasa la llamada al grupo 200.**

Este script atiende las llamadas y pide que se elija una opción, si no se pulsa ninguna opción se pasa la llamada al grupo 200.

Para probar este script se puede llamar al 951 088 536

```
; OpVirtual_2.ivr
; Se descuelga la llamada.
INCOMINGCALL 0
; Reproduce por el canal 0 el fichero Bienvenida.wav
PLAYFILE 0 OpVirtual\Bienvenida 1

:INICIO
; Reproduce por el canal 0 el fichero Opciones.wav, donde le da a elegir a la persona
entre 3 opciones, la reproducción del fichero se para si se pulsa cualquier tecla.
; Si no se pulsa nada en el tiempo previsto, o se pulsa una tecla no válida se pasa
la llamada al grupo 200.
; Opciones.wav "Teclee 1 para Administración, 2 Servicio técnico, 3 para llamar
directamente a una extensión o espere para ser atendido por una operadora.
PLAYFILE 0 OpVirtual\Opciones 0
INPUT 0 0 1 2
PLAYTONE 0 0
IF %v0 == 1
    GOTO OPCION1
ENDIF
IF %v0 == 2
    GOTO OPCION3
ENDIF
IF %v0 == 3
    GOTO OPCION3
ENDIF

CALL2EXTENSION 1 200 20
IFERR 0
    CONNECT 0 1
    EXIT
ENDIFERR

; Si no tiene éxito la llamada al grupo vuelve al inicio.
GOTO INICIO

:OPCION1
CALL2EXTENSION 1 201 20
IFERR 0
    CONNECT 0 1
    EXIT
ENDIFERR
```

; Si no tiene éxito la llamada al grupo vuelve al inicio.

GOTO INICIO

:OPCION2

CALL2EXTENSION 1 202 20

IFERR 0

CONNECT 0 1

EXIT

ENDIFERR

; Si no tiene éxito la llamada al grupo vuelve al inicio.

GOTO INICIO

:OPCION3

; Reproduce por el canal 0 el fichero Extension.wav: "Introduzca el número de extensión".

; Espera que se introduzca el número de extensión ( 3 cifras )

; Si no se pulsa el número de extensión en el tiempo previsto, el número de extensión marcado no es válido o la extensión comunica o no responde, vuelve al INICIO.

PLAYFILE 0 OpVirtual\Extension 0

INPUT 0 1 3 10

PLAYTONE 0 0

CALL2EXTENSION 1 %v1 20

IFERR 0

CONNECT 0 1

EXIT

ENDIFERR

GOTO INICIO

### **4.1.3.- Teclee el número de extensión o espere o pulse 0 para hablar con una operadora.**

Este script atiende las llamadas y pide que se introduzca un número de extensión (3 cifras), si no se pulsa el número de extensión en el tiempo previsto, o se pulsa 0, se pasa la llamada al grupo 200.

Para probar este script se puede llamar al 951 088 537

```
; OpVirtual_3.ivr
; Se descuelga la llamada.
INCOMINGCALL 0
; Reproduce por el canal 0 el fichero Bienvenida.wav
PLAYFILE 0 OpVirtual\Bienvenida 1

:INICIO
; Reproduce por el canal 0 el fichero Extension0.wav, donde se pide que introduzca
el número de extensión al que se quiere llamar..
; Si no se pulsa nada en el tiempo previsto, o se pulsa 0 se pasa la llamada al grupo
200.
; Extension0.wav      "Teclee el número de extensión con la que quiere hablar o
pulse 0 o espere para hablar con una operadora.
    PLAYFILE 0 OpVirtual\Extension0 0
    INPUT 0 0 1 10
    IFERR 1
        GOTO LLAMAROPERADORA
    ENDIFERR
    IF %v0 == 0
        GOTO LLAMAROPERADORA
    ENDIF
    INPUT 0 1 2 10
; Si no pulsa 0 pide el resto de cifras de la extensión.
    GOTO LLAMAREXT

:LLAMAROPERADORA
; Reproduce por el canal 0 el fichero Transfiriendo.wav: "Transfiriendo a una
operadora"
    PLAYFILE 0 OpVirtual\Transfiriendo 1
    CALL2EXTENSION 1 200 10
    IFERR 0
        CONNECT 0 1
        EXIT
    ENDIFERR
; Si no tiene éxito la llamada al grupo vuelve al inicio.
    GOTO INICIO
```

:LLAMAREXT

CALL2EXTENSION 1 %v0 10

IFERR 0

CONNECT 0 1

EXIT

ENDIFERR

; Si no tiene exito la llamada a la extensión marcada vuelve al inicio.

GOTO INICIO



**4.1.4.- "Teclee el número de extensión" con opciones de salida.**

```

INCOMINGCALL 0

:INICIO
PLAYM 0 330 0           ; Reproduce el mensaje.
INPUT 0 0 3 10         ; Se esta reproduciendo el mensaje
mientras.

IF %v0==                ; Si no ha tecleado nada.
    PLAYM 0 334 1       ; "Transfiriendo a una operadora"
    TRANSFER2EXTENSION 210 10
    GOTO INICIO         ; Por si acaso no puede transferir.
ENDIFM

TRANSFER2EXTENSION %v0 10
IFERR 1                  ; Timeout
    PLAYM 0 331
    INPUT 0 1 1 5
IFERR 2                  ; Comunicando
    PLAYM 0 332
    INPUT 0 1 1 5
IFERR 4                  ; Extension inexistente
    PLAYM 0 333 1
    GOTO INICIO
ENDIFERR

IF %v1==#
    RECORDMAILBOX 0 %v0 1
ENDIF

GOTO INICIO

```

Contenido de los mensajes:

330.- "Teclee el número de extensión de la persona con la que quiere hablar o espere para hablar con una operadora".

331.- "La extensión no contesta. Pulse # para dejarle un mensaje o pulse cualquier otra tecla para volver al principio".

332.- "La extensión está ocupada. Pulse # para dejarle un mensaje o pulse cualquier otra tecla para volver al principio".

333.- "El numero de extensión no existe".

334.- "Transfiriendo a una operadora".

---

**4.1.5.- Información asociada al número de teléfono del llamante.**

```
INCOMINGCALL 0
OPENSOCKET 1 127.0.0.1 2000

:INICIO
PLAYM 0 350 0
INPUT 0 0 4 10 ; Pide password.
SEND2SOCKET 1 1 10 %nums %v0 ; Envía el num.tlf y el password.
IF %v1!=
    PLAYM 0 351 1
    PLAYNUMBER 0 %v1 3 1
    HOOK 0
    EXIT
ENDIF
GOTO INICIO
```

Contenido de los mensajes:

350.- "Bienvenido. Teclee su pin".

351.- "Su saldo son".

## **4.2.- Empresas de 80x.**

### **4.2.1.- Carrusel general.**

Para probar este script se puede llamar al 951 088 538

### **4.2.2.- Mailbox general.**

Para probar este script se puede llamar al 951 088 539

### 4.2.3.- Reproducción de mensaje de TAROT pregrabado.

Para probar este script se puede llamar al 951 088 540

```

INCOMINGCALL 0

PLAYM 0 370 1          ; "Coste de la llamada"
PLAYM 0 371 1          ; "Bienvenida al Tarot".

TRANSFER2EXTENSION 20 10 0 A%nums ; La letra indica el servicio.

:ESPERA
PLAYM 0 372 0          ; No hay tarotistas disponibles.
INPUT 0 0 1 5
IFERR 1                ; Si no ha pulsado nada.
    GOTO ESPERA
ENDIFERR

:INICIO
PLAYM 0 373 0          ; "1 para Amor, ....".
INPUT 0 1 1 5
IFERR 1
    GOTO INICIO
ENDIFERR
IF %v1>4                ; Si es mayor que 4, no vale.
    GOTO INICIO
ENDIF
IF %v1==0               ; Si es 0, no vale.
    GOTO INICIO
ENDIF

PLAYM 0 374 2          ; "Estoy barajando cartas"
INPUT 0 0 1 5          ; Espera a recibir una pulsación.

IF %v1==1               ; Amor
    PLAYM 0 376 1
    RANDOM 400 419 2
ENDIF
IF %v1==2               ; Salud
    PLAYM 0 377 1
    RANDOM 420 439 2
ENDIF
IF %v1==3               ; Trabajo
    PLAYM 0 378 1
    RANDOM 440 459 2

```

```

ENDIF
IF %v1==4                                ; Otros temas.
    RANDOM 460 479 2
ENDIF

PLAYM 0 %v2 1                            ; Predicción de las cartas
PLAYM 0 375 1                             ; "¿Quieres otra tirada?"
INPUT 0 0 1 30
IFERR 1                                   ; Si no ha pulsado nada y ha habido
    HOOK 0                                 ; timeout
    EXIT
ENDIFERR
GOTO ESPERA                               ; Vuelta a empezar.

```

Contenido de los mensajes:

370.- "El coste de la llamada es xxx".

371.- "Bienvenido al Tarot-A".

372.- "En este momento no hay ningún tarotista disponible. Pulsa cualquier dígito del teléfono para que nuestro equipo informático te realice una tirada de cartas o continua esperando hasta que alguno quede libre."

373.- "Pulsa 1 si la cuestión que quieres preguntar a las cartas es sobre Amor. Pulsa 2 si es sobre Salud. Pulsa 3 si es sobre Trabajo y Pulsa 4 si es sobre otro asunto."

374.- "Estoy barajando las cartas. Plaf, plaf. Pulsa cualquier dígito de tu teléfono para señalar donde quieres que pare. Plaf, plaf".

375.- "Si quieres saber mas sobre este u otro tema, pulsa una tecla y haremos una nueva tirada".

376.- "Tu tirada es sobre AMOR".

377.- "Tu tirada es sobre SALUD".

378.- "Tu tirada es sobre TRABAJO".

Mensajes 400 a 419: 20 mensajes sobre AMOR.

Mensajes 420 a 439: 20 mensajes sobre SALUD.

Mensajes 440 a 459: 20 mensajes sobre TRABAJO.

Mensajes 460 a 479: 20 mensajes sobre OTROS TEMAS.

---

**4.2.4.- Autenticación de operadores externos.**

```
INCOMINGCALL 0
```

```
:INIO
```

```
PLAYM 0 385 0
```

```
INPUT 0 2 1 10 ; Solicitar tipo de acción.
```

```
IFERR 1
```

```
    GOTO INIO
```

```
ENDIFERR
```

```
IF %v2 == 0
```

```
    GOTO INI1
```

```
ENDIF
```

```
IF %nums ==
```

```
    PLAYM 0 380 1 ; Si se está haciendo un login y no está  
    identificado el tlf, error.
```

```
    HOOK 0
```

```
    EXIT
```

```
ENDIF
```

```
:INI1
```

```
PLAYM 0 381 0
```

```
INPUT 0 0 3 10 ; Solicitar número de operador.
```

```
IFERR 1
```

```
    PLAYM 0 382 1
```

```
    GOTO INI1
```

```
ENDIFERR
```

```
:INI2
```

```
PLAYM 0 383 0
```

```
INPUT 0 1 4 10 ; Solicitar password.
```

```
IFERR 1
```

```
    PLAYM 0 384 1
```

```
    GOTO INI2
```

```
ENDIFERR
```

```
IF %v2 == 0
```

```
    SETEXTENSION %v0 0 %v1 ; Logout
```

```
    PLAYM 0 386 1
```

```
ELSE
```

```
    SETEXTENSION %v0 2 %nums %v1 ; Login
```

```
    IFERR 4
```

```
        PLAYM 0 382 1
```

```
        GOTO INI1
```

```
    IFERR 6
```

```
PLAYM 0 384 1
GOTO INI1
ENDIFERR
PLAYM 0 387 1
ENDIF
HOOK 0
EXIT
```

Contenido de los mensajes:

- 380.- "Número de teléfono sin identificar".
- 381.- "Teclea tu código de operador".
- 382.- "Código de operador incorrecto".
- 383.- "Teclea tu password".
- 384.- "Password incorrecto".
- 385.- "Pulsa 1 para habilitar tu extensión y 0 para deshabilitarla".
- 386.- "Extensión deshabilitada".
- 387.- "Extensión habilitada".

#### 4.2.5.- Pago de visita a web a través de 803/806.

```
INCOMINGCALL 0
```

```
OPENSOCKET 1 127.0.0.1 2000
```

```
PLAYM 0 390
```

```
:INICIO
```

```
PLAYM 0 391 0
```

```
INPUT 0 0 6 20
```

```
IFERR 1 4 5
```

```
    GOTO INICIO
```

```
ENDIFERR
```

```
ONHOOK 0 FIN ; Cuando cuelgue, saltar a FIN
```

```
SEND2SOCKET 1 1 30 O%v0 ; Envía a una aplicación externa "O" y el  
; código leído.
```

```
IF %v1 == 1 ; La aplicación dice "Código inexistente"
```

```
    PLAYM 0 392 1
```

```
    GOTO INICIO
```

```
ENDIF
```

```
PLAYM 0 393 2
```

```
:CICLO
```

```
GOTO CICLO
```

```
:FIN
```

```
SEND2SOCKET 1 1 30 C%v0 ; Envía a una aplicación externa "C" y el  
código leído.
```

```
EXIT
```

Contenido de los mensajes:

390.- "Bienvenido a xxx."

391.- "Teclee el código de 6 cifras que aparece en su navegador."

392.- "Código incorrecto."

393.- "Mantenga el teléfono descolgado, mientras navega por nuestra web."



**4.2.6.- Dedicar canciones.**

```

INCOMINGCALL 0

PLAYM 0 500 1

:INICIO
PLAYM 0 501 0
INPUT 0 0 1 15 ; Pide el autor.
IFERR 1
    GOTO INICIO
ENDIFERR

PLAYFILE 0 A%v0 2
IFERR 4 ; Autor erróneo.
    GOTO INICIO
ENDIFERR

INPUT 0 1 1 30 ; Pide la canción.
IFERR 1
    GOTO INICIO
ENDIFERR

PLAYM 0 502 1 ; La canción escogida es:
PLAYFILE 0 T%v0%v1 1
IFERR 4 ; Canción errónea.
    GOTO INICIO
ENDIFERR

:CONT1
PLAYM 0 503 ; Pulsa 0 ó el número de teléfono.
INPUT 0 2 1 15 ; Pide primer dígito.
IFERR 1
    GOTO CONT1
ENDIFERR
IF %v2==0
    GOTO INICIO
ENDIFERR

INPUT 0 3 8 20 ; Pide los 8 dígitos restantes.
IFERR 1
    GOTO CONT1
ENDIFERR

:CONT2
PLAYM 0 504 1
PLAYNUMBER 0 %v2%v3 4 ; Reproduce el número de tlf. destino.

```

```
PLAYM 0 505 1 ; ¿es correcto?
INPUT 0 4 1 15
IFERR 1
    GOTO CONT2
ENDIFERR
IF %v4==0
    GOTO CONT1
ENDIFERR
```

```
PLAYM 0 506 1
RECORDFILE 0 D%phandle 30
```

```
; última confirmación.
```

```
:CONT3
```

```
PLAYM 0 507
```

```
INPUT 0 4 1 15
```

```
IF %v4==1
```

```
    PLAYM 0 509 1
```

```
    PLAYFILE 0 D%phandle 1 ; Dedicatoria
```

```
    PLAYFILE 0 C%v0%v1 1 ; Canción.
```

```
    GOTO CONT3
```

```
ENDIF
```

```
; Envío de la canción.
```

```
PLAYM 0 508 1
```

```
; Mensaje final.
```

```
HOOK 0
```

```
; Colgar llamada.
```

```
:LLAMADA
```

```
CALL2EXTERIOR 1 %v2%v3 60
```

```
IFERR 1 2 5
```

```
    PAUSE 500
```

```
    GOTO LLAMADA
```

```
ENDIFERR
```

```
PLAYM 1 509 1
```

```
PLAYFILE 1 D%phandle 1
```

```
; Dedicatoria
```

```
PLAYFILE 1 C%v0%v1 1
```

```
; Canción.
```

```
PLAYM 1 510 1
```

```
HOOK 1
```

```
EXIT
```

Contenido de los mensajes:

500.- "Bienvenido a AAAAA. El coste de la llamada es xxx".

501.- "Pulsa 1 para dedicar una canción de David Bustamante. Pulsa 2 para dedicar una canción de Las Tatú".

502.- "La canción que has escogido es".

- 503.- "Pulsa 0 si la canción no es correcta o pulsa el número de teléfono de la persona a la que se lo quieras dedicar".
- 504.- "El número de la persona a la que vas a dedicar la canción es:"
- 505.- "Pulsa 0 si la elección no es correcta o cualquier otra cifra para continuar."
- 506.- "Graba tu dedicatoria al oír la señal. Piiiiii"
- 507.- "Pulsa 1 para escuchar la dedicatoria tal y como la escuchara la persona llamada o cualquier otra cifra para enviarla."
- 508.- "Muchas gracias por tu dedicatoria. La persona que has indicado la recibirá en breve".
- 509.- "Una persona te ha dedicado una canción. Esta es su dedicatoria".
- 510.- "Si quieres dedicar tú otra canción puedes llamar al 906xxxxxx".

Ficheros:

Ficheros por cada autor:

- A1: "Pulsa 1 para xxxx. Pulsa 2 para yyyyy. Pulsa 3 para zzzzzz"
- A2: "Pulsa 1 para aaaa. Pulsa 2 para bbbb."

Ficheros con los títulos de canciones:

- T11: Título de la canción x de David Bustamante.
- T12: Título de la canción y de David Bustamante.
- T13: Título de la canción z de David Bustamante.
- T21: Título de la canción a de Las Tatú
- T22: Título de la canción b de Las Tatú

Ficheros con las canciones:

- C11: Canción x de David Bustamante.
- C12: Canción y de David Bustamante.
- C13: Canción z de David Bustamante.
- C21: Canción a de Las Tatú
- C22: Canción b de Las Tatú

---

**4.2.7.- Validar a un usuario y conectarlo a un salón.**

```
INCOMINGCALL 0
```

```
PLAYM 0 520 1
```

```
:INICIO
```

```
PLAYM 0 521 1
```

```
LOCKGLOBAL 0 30
```

```
SETLOCAL 0 %g0
```

```
INCGLOBAL 0
```

```
RECORDFILE 0 A%v0 30 ; Graba nombre del cliente.
```

```
PLAYM 0 522 2
```

```
:VALIDAR
```

```
CALL2EXTENSION 1 210 30 ; Llama a la extensión 210 para validar.
```

```
IFERR 1 2
```

```
    GOTO VALIDAR
```

```
ENDIFERR
```

```
PLAYM 1 523 1
```

```
PLAYFILE 1 A%v0 1
```

```
INPUT 1 1 1 30 ; Pide número de salón donde meterlo.
```

```
HOOK 1
```

```
IF %v1==0 ; Si 0, volver a pedir nombre.
```

```
    GOTO INICIO
```

```
ENDIF
```

```
CONNECT2ROOM 0 %v1
```

```
EXIT
```

Contenido de los mensajes:

520.- "Bienvenido a L-Party. El coste de la llamada es XXX".

521.- "Di tu nombre al oír la señal. Piiii".

522.- "Espera unos segundos para entrar en una sala de charla".

523.- "El cliente a validar es".

**4.2.8.- Aplicación de tarjetas pre-pago.**

```

INCOMINGCALL 0

OPENSOCKET 1 127.0.0.1 2000

:INICIO
PLAYM 0 530 0
INPUT 0 0 6 10 ; Pide número de tarjeta.
PLAYM 0 531 0
INPUT 0 1 4 10 ; Pide pin.

SEND2SOCKET 1 2 10 A%v0 %v1 ; Envía el número de tarjeta y el pin
IF %v2==A
    PLAYM 0 532 1 ; Números incorrectos.
    GOTO INICIO
ENDIF

PLAYM 0 533 0
PLAYNUMBER 0 %v2 3 1 ; El saldo de su tarjeta es....

:PIDENUM
PLAYM 0 534 0
INPUT 0 3 16 15 # ; Pide el número de teléfono al que
llamar.

SEND2SOCKET 1 2 10 B%v0 %v3 ; Interroga cuantos minutos hay para ese
tlf.
PLAYM 0 535 1
PLAYNUMBER 0 %v2 1 1 ; Tiene disponible xx minutos.
PLAYM 0 536 1

IF %v2==0 ; Si no hay minutos disponibles,
    GOTO PIDENUM
ENDIF

PLAYTONE 0 0
CALL2EXTERIOR 1 %v3 30
IFERR 1
    PLAYM 0 537 1 ; No responde.
    GOTO PIDENUM
IFERR 2
    PLAYM 0 538 1 ; Comunica.
    GOTO PIDENUM
ENDIFERR

ONHOOK 0 FIN0
ONHOOK 1 FIN1

```

```
CONNECT 0 1
SEND2SOCKET 1 2 10 C%v0 %v3 ; Comienzo de la llamada

:CONV
SEND2SOCKET 1 2 10 D%v0 %v3 ; Orden de descontar un minuto.
IF %v2==1
    GOTO FIN1 ; Si no queda saldo.
ENDIF
PAUSE 60
GOTO CONV

:FIN0
ONHOOK 1
HOOK 1
SEND2SOCKET 1 2 10 E%v0 %v3 ; Fin de la llamada
:FIN
EXIT

:FIN1
SEND2SOCKET 1 2 10 E%v0 %v3 ; Fin de la llamada
ONHOOK 0 FIN
PLAYM 0 539 1
PLAYNUMBER 0 %v2 3 1
GOTO INICIO
```

Contenido de los mensajes:

```
530.- "Bienvenido. Teclee su número de tarjeta".
531.- "Teclee su pin".
532.- "Número de tarjeta o pin incorrectos".
533.- "El saldo de su tarjeta es"
534.- "Teclee el número de teléfono al que quiere llamar y pulse
almohadilla"
535.- "Tiene disponible"
536.- "minutos"
537.- "El destino marcado no responde".
538.- "El destino marcado está comunicando".
539.- "El coste de la llamada ha sido de"
```