

BIREME - Centro Latino-Americano e do Caribe de Informação em Ciências da Saúde



Organização Pan-Americana da Saúde

Organização Mundial da Saúde

**WWWISIS: servidor world-wide web para bases de datos ISIS
Version 3.0**

São Paulo, Octubre 1997

Contenido

I. Introducción	3
1. WWWISIS: servidor de bases de datos ISIS para Internet/WWW.....	3
2. WWWISIS y los Common Gateway Interface (CGI) Scripts	3
2.1. El registro virtual del ambiente CGI.....	6
2.1.1. Captura y estructuración de los datos en CGI	7
2.1.2. Generación dinámica de los parámetros de ejecución del WWWISIS.....	11
2.2. El texto de salida de WWWISIS	12
2.2.1. Texto de salida predefinido	12
2.2.2. Texto de salida generado usando el lenguaje de formato del ISIS.....	12
3. WWWISIS y las bases de datos ISIS	15
3.1. Recuperando datos de bases de datos ISIS	16
3.2. Actualizando los datos en bases de datos ISIS.....	16
3.3. Campos de información de proceso del WWWISIS.....	17
4. Bases de datos log del WWWISIS y operación sin estados del cliente-servidor.....	17
5. Procedimientos/scripts LOADISO y FULLINV	18
II. Parametrización del WWWISIS.....	19
1. Versión de WWWISIS, sintaxis general, resumen de parámetros	19
1.1. Menú de opciones de parámetros	19
1.2. Version de WWWISIS	19
1.3. Sintaxis general de parámetros.....	20
1.4. Parámetros obligatorios	21
1.5. Parámetro args y mensajes de error de WWWISIS.....	21
1.6. Grupos de parámetros	22
2. Parámetros generales	23
3. Parámetros relacionados con expresiones booleanas.....	24
4. Parámetros relacionados con rango alfabético de términos	26
5. Parámetros relacionados con rango de registros del archivo maestro.....	27
6. Parámetros relacionados con el registro virtual de ambiente CGI	28
7. Parámetros relacionados con la actualización de una base de datos.....	29
8. Tags de campos del registro log	31
9. Códigos de retorno WWWISIS.....	32
10. Parámetros reservados	33
III. Manual de referencia de parámetros WWWISIS	34

I. Introducción

1. WWWISIS: servidor de bases de datos ISIS para Internet/WWW

WWWISIS es un sistema desarrollado y distribuido por BIREME/PAHO/WHO, especialmente diseñado para actuar como servidor para bases de datos ISIS en un ambiente cliente/servidor WWW. Proporciona funciones para operaciones de búsqueda y entrada de datos en bases de datos ISIS.

WWWISIS opera como servidor a través de la WWW Common Gateway Interface (CGI). Las operaciones reales son dirigidas por parámetros. Ejemplos de parámetros son: el nombre de la base de datos de entrada, una búsqueda booleana y un formato de salida. Una característica notable es su capacidad para generar dinámicamente los parámetros de ejecución, basado en los datos enviados al proceso CGI desde el visualizador WWW.

WWWISIS puede ser llamado directamente o como una línea de comando dentro de un script CGI. Un script CGI puede contener una o más llamadas al WWWISIS. Además, puede llamarse a sí mismo varias veces.

Los lenguajes de búsqueda, formato y actualización de campos del ISIS están disponibles. Las especificaciones de búsqueda pueden ser capturadas usando formas WWW, pasadas al WWWISIS vía CGI y los resultados formateados y entregados al cliente. De la misma forma, es posible implementar operaciones de actualización de bases de datos. Por defecto, WWWISIS presupone un ambiente multiusuario.

La salida del WWWISIS para el visualizador WWW es generada usando el lenguaje de formato del ISIS. Teniendo en cuenta la posibilidad de incluir códigos HTML en las especificaciones de formato, usando literales condicionales e incondicionales, es posible el desarrollo de interfaces de búsqueda cliente/servidor totalmente gráficas para bases de datos ISIS (en un ambiente Internet o Intranet).

El código fuente de WWWISIS usa la Biblioteca Interface Cisis, desarrollada por BIREME. Corriendo en DOS, el módulo ejecutable es totalmente compatible con la estructura de archivo UNESCO CDS/ISIS 3.07. Sin embargo, en sistemas operativos UNIX no existen garantías de esta compatibilidad.

WWWISIS es distribuido con el procedimiento LOADISO para crear archivos maestros a partir de archivos ISO y el procedimiento FULLINV para crear archivos invertidos. Se recomienda que en el sistema operativo UNIX, el archivo maestro de la base de datos sea generado a partir de un archivo ISO usando el procedimiento LOADISO y el (los) archivo(s) invertido(s) generado(s) totalmente usando el procedimiento FULLINV.

2. WWWISIS y los Common Gateway Interface (CGI) Scripts

WWWISIS puede llamarse directamente digitando una línea de comando del sistema operativo, por ejemplo:

```
wwwisis db=cds "bool=water and plant" pft=@cds.pft
```

Este método es particularmente útil para conocer y probar los parámetros del WWWISIS.

Sin embargo, es a través del WWW CGI que WWWISIS actúa como servidor de bases de datos ISIS para Internet/WWW.

CGI es una parte integral de los servidores WWW que provee un mecanismo para llamar otros programas. De este modo, las solicitudes de un visualizador WWW a un servidor WWW pueden ser enviadas a un programa o script, que a su vez puede recuperar datos de una base de datos y enviar la salida al servidor WWW, que la transfiere al visualizador WWW, generalmente en formato HTML.

Los programas o scripts CGI residen en uno o más directorios reconocidos por el servidor WWW como rutas de acceso CGI. Las rutas de acceso CGI son configuradas por el administrador del servidor WWW (generalmente, el nombre del subdirectorio CGI es */cgi-bin/* bajo el directorio raíz del servidor, pero puede ser mapeado para cualquier otra ruta de acceso). Por tanto, todos los programas o scripts que serán llamados usando CGI deben residir en las correspondientes rutas de acceso CGI.

La forma más común de capturar datos en visualizadores WWW es a través de formas. FORM es un elemento del lenguaje HTML, que permite la especificación de elementos de Entrada, como field text, list box, check box, radio box, etc. Una búsqueda Isis, por ejemplo, puede digitarse en un elemento de Entrada de tipo *text* y los límites de la búsqueda pueden capturarse usando elementos de tipo *list box*, *check box* o *radio box*.

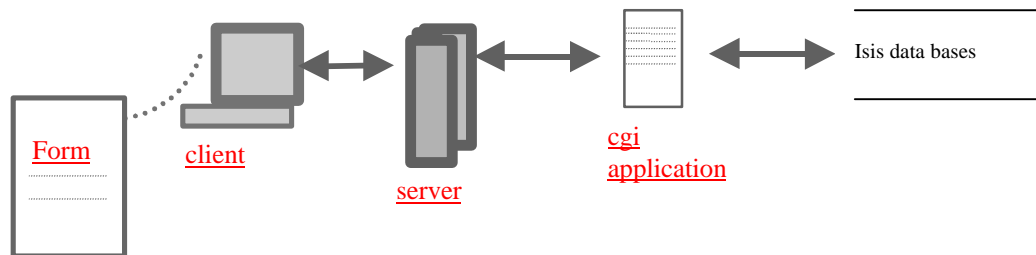
Cada elemento de Entrada en un FORM tiene dos atributos básicos asociados: el *nombre* que identifica al elemento de Entrada y el *valor* asignado al mismo.

Un atributo clave del elemento FORM es ACTION, que permite especificar el nombre del programa o script a activar vía CGI. A continuación se muestra una especificación de forma en lenguaje HTML:

```
<html>
.
.
Enter search:<br>
<form action="http://cgi-bin/isisSearch" method="POST">
<input type=text name="boolean" value="">
<input type=submit name="button" value="pesquisa">
</form>
.
.
</html>
```

El primer elemento de Entrada usado en el ejemplo anterior de forma tiene atributo *type* igual a *text*, seguido por el atributo *name* igual a *boolean* y el atributo *value* igual a null. Luego de ser digitada en el campo de entrada de la forma, la expresión de búsqueda será almacenada en el atributo *value*. El tipo del segundo elemento de Entrada es un botón *submit* que cuando se clicla envía los datos introducidos en el campo *boolean* hacia el servidor WWW.

El uso de FORMas en la parte del cliente y aplicaciones CGI en la del servidor comprende los mecanismos claves para capturar solicitudes de búsqueda y/o entrada de datos y enviarlas a una base de datos. El siguiente diagrama describe cómo operan estos mecanismos:



Resumiendo, el visualizador WWW (cliente) captura los datos a través de formas y los envía al servidor WWW, que llama la aplicación CGI (script o programa) especificada en el atributo ACTION del elemento FORM. La aplicación CGI envía su salida al servidor WWW, que la transfiere al visualizador WWW. Este intercambio de datos cliente-servidor generalmente se repite varias veces permitiendo la interacción entre un usuario y una base de datos. Por ejemplo: (a) una búsqueda puede refinarse mediante interacciones sucesivas; (b) los resultados de una búsqueda pueden presentarse en subconjuntos sucesivos; o (c) una operación de entrada de datos puede atravesar varios pasos cuando las reglas de consistencia lo requieran. El uso de texto oculto en FORMas HTML es una forma común de llevar los datos de un paso al próximo.

Existen dos métodos básicos que el WWW usa para transferir los datos del cliente a la aplicación CGI: GET y POST. El método a utilizar es especificado en el atributo METHOD del elemento HTML FORM. GET es el método por defecto.

Con el método GET, el visualizador envía al servidor una cadena de caracteres que incluye la dirección del cliente en formato Uniform Resource Locator (URL), seguida por el carácter "?" y por una cadena que contiene todos los datos capturados en la forma (nombres y valores de los elementos de entrada); cuando el servidor recibe esta cadena, separa la subcadena que está luego del carácter "?" y se la asigna a una variable de ambiente llamada QUERY_STRING, que puede ser usada por la aplicación CGI.

Con el método POST, el servidor envía la cadena de datos al CGI a través de la Entrada Estándar del sistema operativo, que es luego leída por la aplicación CGI. Es preferible usar el método POST para procesos que involucran grandes volúmenes de datos.

WWWISIS puede ser activado en un proceso CGI de cinco formas:

- a. Llamando al módulo ejecutable directamente desde la caja de diálogo de apertura de URL del visualizador WWW, por ejemplo:

```
http://www.xxx.xx/cgi-bin/wwwisis.exe/[hello]/
```

Este llamado presupone que WWWISIS esté almacenado en el directorio *cgi-bin*. Este llamado es el llamado estándar para chequear si WWWISIS está instalado correctamente y en condiciones de responder.

- b. Llamando al módulo ejecutable directamente desde una especificación del elemento *anchor* de HTML, como argumento del atributo *href*, por ejemplo:

```
<a href="/cgi-bin/wwwisis.exe/[menu=6]/"> WWWISIS error codes list </a>
```

```
<a href="/cgi-bin/wwwisis.exe/[in=param.in]/">WWWISIS via parameter file</a>
```

- c. Llamando al módulo directamente desde una especificación del elemento *form* de HTML, como argumento del atributo *action*, por ejemplo:

```
<form method=get action="/cgi-bin/wwwisis.exe/[menu=6]/"> ..... </form>
```

```
<form method=post action="/cgi-bin/wwwisis.exe/[in=param.in]/"> ... </form>
```

- d. Como línea de comando de un script especificado a través del elemento *anchor* de HTML, como argumento del atributo *href*, por ejemplo:

```
<a href="/cgi-bin/menu6.bat"> WWWISIS error codes list </a>
```

```
<a href="/cgi-bin/param.sh">WWWISIS via parameter file</a>
```

- e. Como una línea de comando de un script especificado mediante el elemento *form* de HTML, como argumento del atributo *action*, por ejemplo:

```
<form method=get action="/cgi-bin/Mibatch.bat"> ..... </form>
```

```
<form method=post action="/cgi-bin/Miscript.sh"> ... </form>
```

Los métodos b y c son generalmente la forma preferible de llamar al WWWISIS en Windows NT, cuando se está operando con el Microsoft Internet Information Server. Todos los métodos se utilizan en el sistema operativo UNIX.

Note que sólo los siguientes parámetros pueden ser colocados a continuación del módulo de ejecución: *args*, *ctttype*, *hello*, *in* y *menu*.

2.1. El registro virtual del ambiente CGI

Cuando WWWISIS comienza, crea primero en memoria un registro de archivo maestro ISIS llamado registro virtual de ambiente CGI. WWWISIS usa este registro para almacenar datos enviados por el servidor WWW al proceso CGI.

WWWISIS procesa los datos transferidos por el servidor WWW, tanto en el método GET como en el POST, a través del parámetro *cgi*. Por tanto, no es necesario preocuparse con la manipulación de los datos enviados por el servidor a la aplicación CGI. En general, aplicaciones WWWISIS simples y complejas no requieren programas adicionales o

comandos de sistemas operativos; esta facilidad permite que una aplicación WWWISIS pueda ser fácilmente transferida de una plataforma a otra; en particular, es posible desarrollar una aplicación entera en una PC individual y luego transferirla a un servidor en red.

El registro virtual del ambiente CGI puede ser accedido y formateado a través de los parámetros *cgi*, *cmd*, *stop* y *prolog*.

2.1.1. Captura y estructuración de los datos en CGI

Cuando el parámetro *cgi* está presente, WWWISIS reúne los datos enviados por el servidor, si hay alguno, y los almacena en el registro virtual de ambiente CGI. WWWISIS espera que los datos enviados por el servidor estén estructurados en pares de elementos de datos: el nombre (*name*) del campo de entrada y el valor (*value*) correspondiente. Estos pares corresponden a los atributos *name* y *value* del elemento de Entrada de un FORM HTML, como en el ejemplo siguiente:

```
....
<form action="http://cgi-bin/getmfn" method=post>
  <input type=hidden name="mfn" value="50">
  <input type=submit name="button" value="Get mfn 50">
</form>
....
```

WWWISIS almacena los pares de datos del elemento *input* en el registro virtual de ambiente CGI, siguiendo un esquema de asignación de tags poderoso y flexible, que ofrece diferentes métodos para establecer una correspondencia entre el atributo *name* y el número de tag. El uso de un determinado método depende de cómo serán tratados los datos en el proceso CGI.

Los métodos usados por WWWISIS para reunir y asignar tags a los datos enviados al servidor por el cliente son descritos a continuación:

a. *tcg*=<tag>

El parámetro *tcg* permite la especificación del número de tag donde serán almacenados los datos en el registro virtual y ambiente CGI. El componente nombre del elemento de entrada se almacena en el subcampo *^n* y el valor del elemento de entrada en el subcampo *^v*. Por ejemplo:

```
wwwisis tcg=2001 cgi=@genparm.cgi
```

El tag especificado en el parámetro *tcg* es repetitivo por defecto.

Si no se especifica el parámetro *tcg*, los pares *name* y *value* son estructurados y almacenados en el campo de datos con tag 100, subcampos *^n* y *^v*. Cada par se almacena en una ocurrencia diferente del campo 100. Así, en el ejemplo,

```

....
<form action="http://cgi-bin/getmfn" method=post>
    <input type=hidden name="mfn" value="50">
    <input type=submit name="button" value="Get mfn 50">
</form>
....

```

El campo 100 del registro virtual de ambiente CGI contendrá dos ocurrencias:

Tag	Contenido
100	^nmfn^v50
100	^nbutton^vGet mfn 50

El mismo grupo de datos y la acción podrían ser codificados usando el atributo *href* del elemento *anchor*, por ejemplo:

```

....
<a href="http://cgi-bin/getmfn?mfn=50">Get mfn 50</a>
....

```

b. *pfxtag*=<prefijo>

El uso del parámetro *pfxtag* permite asignar tags individuales a los datos recibidos, según la correspondencia entre el argumento *pfxtag* y el contenido del atributo *name* del elemento de Entrada: si *name* contiene el mismo prefijo especificado en el parámetro *pfxtag*, seguido por un número, este será el número de tag en el que será almacenado *value* en el registro virtual de ambiente CGI. A continuación se muestra un ejemplo de código HTML usando el elemento FORM, usando identificaciones de tag como prefijo en los atributos *name*:

```

....
<form action="http://cgi-bin/frmproc.sh" method=post>
    Nombre: <input type=text name="tag1"><br>
    Direccion: <input type=text name="tag2">
    <input type=submit name="boton" value="Envia" >
</form>
....

```

En este ejemplo, suponiendo que el usuario introduzca los siguientes datos en los campos de entrada:

Nombre: Joao da Silva
 Direccion: Rua John Smith, 300

cuando clique el botón *Envia*, los datos de entrada serán enviados al servidor WWW, que los transferirá a la Entrada Estándar del script CGI *frmproc.sh*. Si este script incluye un llamado al WWWISIS con los parámetros *pfxtag* y *cgi*, como en el ejemplo,


```
wwwisis pfxitag=tag cgi=@cgifmt.cgi
```

los datos enviados serán almacenados en el registro virtual de ambiente CGI según la siguiente estructura:

Tag	Contenido
1	Joao da Silva
2	R. John Smith, 300
100	^nboton^vEnvia

Note que cuando una subcadena *name* coincide con el argumento *pfxitag*, sólo el contenido real de *value* es almacenado en el tag del campo de datos correspondiente.

Si el prefijo no coincide con el argumento *pfxitag*, los datos son almacenados en el campo con tag 100 o en el argumento *tcg*, si está presente.

c. Subcampos `^t<tag>` y `^r<código-ascii>`

Si *name* contiene el identificador de subcampo `^t` seguido por un número, este será el número del tag del campo de datos en el que *value* será almacenado en el registro virtual de ambiente CGI. Para separar diferentes ocurrencias durante la entrada de datos, es posible especificar en el *name* el identificador de subcampo `^r` seguido por un número que representa el código ASCII del separador repetitivo. A continuación se muestra un ejemplo de código HTML usando el elemento Form, usando identificación de tag `^t` e indicador repetitivo `^r` en los atributos *name*:

```
....  
<form action="http://cgi-bin/frmproc.sh" method=post>  
  Nombres: <input type=text name="nom^t1^r13"><br>  
  Direccion: <input type=text name="dir^t2">  
    <input type=submit name="boton" value="Envia" >  
</form>  
....
```

En este ejemplo, suponiendo que el usuario introduzca los siguientes datos en los campos de entrada:

```
Nombres:   Joao da Silva  
           Antonio Padova  
Direccion: Rua John Smith, 300
```

Cuando clique el botón *Envia*, los datos de entrada serán enviados al servidor WWW, que los transferirá a la entrada estándar del script CGI *frmproc.sh* que incluye un llamado al WWWISIS, como en el ejemplo:

```
wwwisis cgi=@cgifmt.cgi
```

los datos enviados serán almacenados en el registro virtual de ambiente CGI según la siguiente estructura:

Tag	Contenido
1	Joao da Silva
1	Antonio Padova
2	R. John Smith, 300
100	^nboton^vEnvia

Note que en el ejemplo anterior, el separador repetitivo es *new line* .

d. Subcampos ^w y ^r

Este método está orientado especialmente para la entrada de datos, o sea, crear un nuevo registro o modificar un registro existente. (Note que los otros métodos pueden también ser usados para entrada de datos cuando son combinados con el parámetro *proc*.)

Cuando sólo está presente el indicador de subcampo ^w en el argumento del atributo *name*, los datos reunidos no son almacenados en el registro virtual de ambiente CGI si no que van directamente para el registro de la base de datos que está siendo procesado. Cuando también se usan el esquema prefijado o el subcampo ^t , los datos reunidos son almacenados tanto en el registro virtual de ambiente CGI como en el registro de la base de datos que está siendo procesado. El ^r puede ser usado para especificar el separator repetitivo. El indicador ^w se usa de tres formas:

- (i) puede colocarse un número de tag a continuación especificando el tag donde serán almacenados los datos en el registro de la base de datos, por ejemplo:

```
....  
<form action="http://cgi-bin/frmpoc" method=post>  
  Author: <input type=text name="^w10^r37"><br>  
  <input type=submit name="boton" value="Envia" >  
</form>  
....
```

En el ejemplo anterior, los nombres de los autores serán almacenados directamente en el campo de tag 10 del registro de la base de datos que está siendo creado o modificado.

- (ii) puede ser codificado junto con un número de tag prefijado, por ejemplo:

```
....  
<form action="http://cgi-bin/frmpoc" method=post>  
  Author: <input type=text name="tag10^r37^w"><br>  
  <input type=submit name="boton" value="Envia" >  
</form>  
....
```

En el ejemplo anterior, los nombres de los autores serán almacenados en el campo de tag 10 tanto del registro virtual de ambiente CGI como del registro de la base de datos que está siendo creado o modificado.

(iii) puede codificarse junto con el indicador de subcampo \w , por ejemplo:

```
....  
<form action="http://cgi-bin/frmproc" method=post>  
  Author: <input type=text name="Au^t10^r37^w500"><br>  
  <input type=submit name="boton" value="Envia" >  
</form>  
....
```

En el ejemplo anterior, los nombres de los autores serán almacenados en el campo de tag 10 del registro virtual de ambiente CGI y en el campo de tag 500 del registro de la base de datos que está siendo creado o modificado.

Resumiendo, la diferencia entre \w y \w es sólo el registro donde los datos reunidos serán almacenados. Usando \w , los datos son almacenados en el registro virtual de ambiente CGI. Usando \w , los datos son almacenados en el registro recuperado de la base de datos o en el nuevo registro que está siendo introducido. Usando ambos, los datos son almacenados en ambos registros.

e. Agregando variables de ambiente CGI

Otra forma de agregar datos al registro virtual de ambiente CGI es a través del parámetro *env*, que permite agregar el valor de variables de ambiente disponibles durante el procesamiento del script CGI. El siguiente ejemplo agregará las variables *SERVER_NAME* y *HTTP_USR_AGENT*, que identifican respectivamente el nombre del servidor y el cliente visualizador, al registro virtual de ambiente CGI:

```
wwwisis env=SERVER_NAME=51,HTTP_USR_AGENT=52 cgi=@cgifmt.cgi
```

El registro virtual de ambiente CGI puede ser formateado usando los parámetros *cgi*, *cmd*, *prolog* y *stop*. Estos parámetros son activados solo al inicio de una ejecución del WWWISIS.

2.1.2. Generación dinámica de los parámetros de ejecución del WWWISIS

La operación del WWWISIS es dirigida por parámetros. Una característica importante del WWWISIS es su capacidad de auto-generación del contenido de los parámetros de ejecución.

El parámetro *cgi* se usa para generar los parámetros de ejecución del WWWISIS. Su argumento contiene una especificación de formato que es ejecutada sobre el registro virtual de ambiente CGI, siendo interpretada cada línea de salida como parámetro de ejecución por el WWWISIS. En otras palabras, **los parámetros del WWWISIS pueden ser generados**

dinámicamente ("al vuelo") o podemos considerar que la presencia y los argumentos del parámetro de ejecución pueden ser "programados".

Lo siguiente es un llamado a WWWISIS usando el parámetro *cgi*:

```
wwwisis cgi=@genparm.cgi
```

El parámetro *cgi* especifica el fichero *genparm.cgi* que contiene una especificación de formato que generará los parámetros reales de ejecución, uno por línea. Un ejemplo de este formato puede ser:

```
'db=cds',/  
'bool=plant and water'/,  
'pft=@cds.pft'
```

En el siguiente ejemplo, el fichero *genparm.cgi* formatea el campo 100 para recibir la expresión booleana introducida a través de un elemento de Entrada de un FORM WWW llamado *boolean*:

```
'db=cds',/  
'bool=',(if v100^n = 'boolean' then v100^v fi)/,  
'pft=@cds.pft'
```

Note que el formato presupone que el campo 100 sea repetitivo, cada ocurrencia con subcampos nombre (^n) y valor (^v).

El parámetro *cmd* es similar a *cgi*, excepto que **no agrega datos en el registro de ambiente CGI**. No puede usarse junto con *cgi* en la misma corrida de WWWISIS.

2.2. El texto de salida de WWWISIS

WWWISIS produce como salida datos que están pre-definidos internamente, pero en la mayoría de los casos se usa el Lenguaje de Formato del ISIS para generar textos de salida "al vuelo".

2.2.1. Texto de salida predefinido

WWWISIS provee parámetros para la salida de textos internos pre-definidos, incluyendo menús de los parámetros disponibles, un mensaje de saludo para prueba y los diferentes encabezados de ficheros http. A continuación se muestra un ejemplo de estos parámetros:

```
wwwisis hello
```

```
wwwisis ctttype=text/plain menu=6
```

2.2.2. Texto de salida generado usando el lenguaje de formato del ISIS

WWWISIS pasa los datos de salida para el servidor WWW usando los parámetros *stop*, *prolog*, *pft* y *epilog*. Todos estos parámetros contienen especificaciones de formato ISIS.

En un llamado a WWWISIS, el parámetro *stop* da formato al registro virtual de ambiente CGI. **Actúa como un semáforo**. Si el formato *stop* genera alguna salida, esta es enviada directamente a la salida estándar y **WWWISIS detiene la ejecución**. Es particularmente útil para el procesamiento de varios preámbulos, como login de usuario y password, disponibilidad del sistema, consistencia en la entrada de datos, etc.

El *prolog* da formato al registro virtual de ambiente CGI y se usa generalmente para producir el encabezamiento del texto de salida, por ejemplo:

```
wwwisis db=cds prolog=@header.pft from=1 pft=@cds.pft  
epilog=@tail.pft
```

donde *header.pft* contiene la siguiente especificación de formato:

```
'<html> <body>'
```

El *pft* da formato a los registros recuperados de base de datos de entrada y es usado generalmente para producir el cuerpo del texto de salida.

El *epilog* da formato al registro log del WWWISIS y puede usarse para producir el final del texto. En el ejemplo anterior, *tail.pft* podría contener:

```
'</body></html>'
```

La tabla siguiente describe para cada parametro que espera una especificación de formato, el registro fuente y el significado de la salida :

Parámetro	Registro fuente	Salida
stop	Registro virtual de ambiente CGI	Salida estándar. Si se produce alguna salida, WWWISIS detiene la ejecución.
prolog	Registro virtual de ambiente CGI	Salida estándar
cgi	Registro virtual de ambiente CGI tag de campo 100 por defecto	Parámetros de ejecución del WWWISIS. Un parámetro por línea
cmd	Registro virtual de ambiente CGI sin tag de campo 100 por defecto	Parámetros de ejecución del WWWISIS . Un parámetro por línea
proc	Registro recuperado de la base de datos especificado en el parámetro <i>db</i>	Especificación Cisis de actualización de campo
pft	Registro recuperado de la base de datos especificado en el parámetro <i>db</i>	Salida estándar
pfttag, pftgen	Registro recuperado de la base de datos especificado en el parámetro <i>db</i>	Argumento <i>pft</i> real
epilog	Registro LOG virtual	Salida Estándar
next	Registro LOG virtual	Parámetros de ejecución del WWWISIS para el próximo llamado recursivo. Un parámetro por línea.

Como la especificación de formato puede usar los comandos *ref* (reference) y *l* (lookup), es posible, en cualquiera de los parámetros anteriores, acceder diferentes bases de datos además de la especificada en el parámetro *db*. Puede darse a los parámetros *cmd*, *cgi*, *pft*, *prolog* y *epilog* el valor *ALL* para obtener todos los campos de datos del registro, por ejemplo:

```
wwwisis db=cds bool=plant pft=@cds.htm epilog=ALL
```

Para hacer que una especificación de formato sea nula, use una *coma* como especificación de formato, por ejemplo:

```
wwwisis db=cds from=1 count=1 pft=, log=x
```

La Biblioteca Interface Cisis usada por WWWISIS introduce varias extensiones a la Lenguaje de Formato ISIS estándar, que pueden ser usadas en cualquiera de los parámetros anteriores (ver Anexo 1).

El texto de salida enviado al servidor es generalmente codificado en HTML para hacer uso de todas sus facilidades gráficas y de hipertexto. La salida es precedida por un encabezamiento que usualmente contiene una especificación de tipo de contenido terminada por una línea en blanco. Los tipos de contenido comunmente usados en las aplicaciones WWWISIS son *text/html* y *text/plain*, el primero indicando que sigue un texto HTML, y el segundo que sigue un texto sin marcación. Un ejemplo de una salida de un CGI es:

```
Content-type: text/html
```

```
<html>
<head><title>Example of cgi-output</title></head>
<body>
<h1> WWWISIS: the way to Internet Isis</h1>
.
.
</body>
</html>
```

El ejemplo anterior de salida de CGI pudiera ser producido por especificaciones de formato usando literales: las 6 primeras líneas podrían ser la salida a través del parámetro *prolog* y las dos últimas líneas a través del parámetro *epilog*. El siguiente script implementa este ejemplo:

```
#!/bin/ksh
echo Content-type:text/html
echo
wwwisis cgi=@genparm.cgi
```

El archivo *genparm.cgi* podría contener la siguiente especificación de formato:

```
'db=cds' /
'prolog=@header.pro' /
'pft=@cads.pft' /
'from=1' /
'count=10' /
'epilog=@tail.epi'
```

donde *header.pro* contiene:

```
'<html>',
'<head><title>Example of cgi -output</title></head>',
'<body>',
'<h1> WWWISIS: the way to Internet Isis</h1>',
```

y *tail.epi* contiene:

```
'</body>',
'</html>',
```

3. WWWISIS y las bases de datos ISIS

WWWISIS recupera datos de una base de datos ISIS y puede actualizarla.

3.1. Recuperando datos de bases de datos ISIS

Existen tres formas básicas para recuperar datos de una base de datos:

(a) una *expresión booleana* (parámetro *bool*), que es aplicada a los términos de un archivo invertido y produce, como resultado, un conjunto de registros del archivo maestro que satisfacen la expresión booleana; el siguiente ejemplo muestra el uso de una expresión booleana para recuperar registros:

```
wwwisis db=cds "bool=plant and water" pft=@cds.pft
```

(b) un rango alfabético de términos del archivo invertido (parámetros *k1* y *k2*); el siguiente ejemplo muestra los términos de recuperación con raíz igual a *pla*:

```
wwwisis db=cds k1=pla k2=plazzzzzzz "pft=v1,c31,v2/"
```

(c) un rango secuencial de registros del archivo (parámetros *from* y *to*); el siguiente ejemplo muestra la recuperación de los 10 primeros registros:

```
wwwisis db=cds from=1 to=10 pft=@cds.pft
```

WWWISIS opera sobre un conjunto de registros ISIS recuperados usando uno o una combinación de estos métodos. La operación consiste en recorrer el conjunto de registros recuperados, aplicando en cada registro el proceso diferente especificado por los parámetros. El proceso final en cada registro es siempre el formateo del registro en la salida estándar a través de los parámetros *pft*, *pfttag* o *pftgen*.

3.2. Actualizando los datos en bases de datos ISIS

WWWISIS permite también la actualización de una base de datos. Es posible crear nuevos registros (parámetro *new*), actualizar registros existentes (parámetros *read* y *write*) y actualizar el archivo invertido (parámetro *fst*).

WWWISIS ofrece dos parámetros básicos para manipular el contenido de los registros del archivo maestro ISIS: el parámetro *proc* para agregar y borrar campos y el parámetro *gizmo* para el reemplazo de cadenas. La actualización de un archivo invertido para reflejar el contenido de registros nuevos y modificados es activada con el parámetro *fst*. Se asume que el proceso de actualización corra siempre en multi-usuario y en un medio ambiente cliente-servidor sin estados. WWWISIS incorpora mecanismos para bloquear y desbloquear registros y bases de datos y asegurar así la integridad del proceso de actualización.

El siguiente ejemplo muestra un llamado al WWWISIS para crear un nuevo registro:

```
wwwisis db=cds new proc=@newrec.pft pft=@cds.pft
```

Los campos de datos del nuevo registro son agregados a través de la especificación de actualización de campos en el parámetro *proc*.

3.3. Campos de información de proceso del WWWISIS

Para facilitar la generación de la salida formateada usando los parámetros *pft*, *pfttag* o *pftgen*, WWWISIS agrega a cada registro que está siendo procesado varios campos de datos con el contenido de los parámetros usados en el llamado al WWWISIS así como los resultados del proceso realizado al registro. Estos campos se denominan **campos de información de proceso**.

Por defecto, estos campos de datos tienen tags numerados de 1000 en adelante, como por ejemplo, el campo de tag 1021 contiene la expresión booleana, el campo 1002 contiene el número total de registros recuperados, etc. Es posible, sin embargo, cambiar estos tags en caso de que estén presentes en los registros de entrada de la base de datos. Los parámetros *t-*, como por ejemplo, *tbo*, *ttt*, etc. son usados para este fin.

Ejemplo:

```
wwwisis tcg=2001 cgi=@genparm
```

en este caso los datos CGI serán almacenados en el campo de tag 2001, o sea, la especificación de formato almacenada en el fichero *genparm* debe referirse al campo de tag 2001 en lugar de al campo de tag 100.

Para conocer los campos de información de proceso disponibles en un llamado al WWWISIS, use la palabra clave *ALL* en el parámetro *pft*, como por ejemplo:

```
wwwisis db=cds bool=water pft=ALL
```

Note que **los campos de información de proceso sólo están disponibles para dar formato a los registros**. Así, cuando se crea un registro usando *new* o se actualiza usando el parámetro *write*, estos campos de información no son grabados en la base de datos.

4. Base de datos log del WWWISIS y operación sin estados del cliente-servidor

Al final de cada corrida, WWWISIS crea en memoria un registro log que contiene los datos relacionados con el proceso.

Por defecto, estos datos son almacenados en los campos con tags de 1000 en adelante. Contienen todos los parámetros usados en la ejecución así como el resultado del proceso. Por ejemplo, el campo 1091 contiene el código de retorno.

El registro log puede ser formateado a través del parámetro *epilog* y/o salvado en una base de datos usando el parámetro *log*.

Para conocer los campos log disponibles al final de un llamado al WWWISIS, use la palabra clave *ALL* en el parámetro *epilog*, como por ejemplo:

```
wwwisis db=cds bool=water pft=, epilog=ALL
```

La operación cliente-servidor WWW sigue el *modo sin estados*, o sea, un servidor WWW no almacena ningún dato relacionado con la aplicación CGI cuando termina. WWWISIS tampoco graba automáticamente ninguna información acerca de un proceso. Sin embargo, WWWISIS permite a través del parámetro *log* la creación de una base de datos de salida ISIS que tiene un solo registro - el registro log WWWISIS. Ejemplo:

```
wwwisis db=cds bool=water pft=, epillog=ALL log=cdslog
```

Además de la base de datos log WWWISIS, es responsabilidad de la aplicación CGI dar tratamiento a los estados intermedios si fuera necesario, lo cual puede realizarse transfiriendo los datos de estado al cliente, generalmente a través de los campos ocultos del HTML FORM, o almacenarlo en el computador del servidor, lo cual demandará mecanismos para recuperar lo almacenado.

5. Procedimientos/scripts LOADISO y FULLINV

Los procedimientos/scripts LOADISO y FULLINV son distribuidos junto al WWWISIS. Estos procedimientos se construyen usando los utilitarios *mx* e *ifload*, que son parte del paquete de **Utilitarios CISIS** desarrollado y distribuido por BIREME. Para sistemas operativos DOS, estos procedimientos tienen extensión *bat* y para UNIX, la extensión es *sh*.

LOADISO puede usarse para cargar un archivo ISO ISIS en un archivo maestro. Es llamado como una línea de comando con dos parámetros: el nombre y extensión del archivo iso y el nombre del archivo maestro ISIS, por ejemplo:

```
loadiso.sh /bases/cds/cds.iso cds
```

FULLINV puede usarse para generar totalmente un archivo invertido a partir de un archivo maestro. Se llama como una línea de comando con cuatro parámetros, los tres primeros obligatorios: el nombre del archivo maestro, la Tabla de Selección de Campos (*fst*), el nombre del archivo invertido y una bandera (*-reset*) cuya presencia retiene los indicadores pendientes del archivo invertido del archivo maestro. Ejemplo:

```
fullinv.sh /bases/cds/cds /bases/cds/cds.fst /bases/cds/cds
```

Estos procedimientos deben ser adaptados a la configuración local.

II. Parametrización del WWWISIS

La operación del WWWISIS es dirigida por parámetros. Así, el desarrollo de cualquier aplicación WWWISIS consiste en la "parametrización" o "programación" de sus parámetros de operación. Esta sección discute la organización y el uso de los parámetros del WWWISIS. En la **Section III** se da una descripción detallada de cada parámetro individual.

Cuando se invoca en la línea de comando sin ningún parámetro, WWWISIS emite una secuencia de pantallas de texto listando todos los parámetros posibles. Estas pantallas actúan como sistema interno de ayuda, reflejando las funciones, parámetros y opciones.

Esta sección de documentación tratará de cada uno de los grupos de opciones y parámetros del WWWISIS, siguiendo la estructura general listada en las pantallas de ayuda internas.

1. Versión de WWWISIS, sintaxis general, resumen de parámetros

1.1. Menú de opciones de parámetro

Pantalla 0:

```
CISIS Interface v3.33/PC/M/30000/10/30/I - WWWIisis 3.0
CISIS Interface v3.33/4GB/GIZ/DEC/FAT/CIP
Copyright (c) BIREME/PAHO 1997. All rights reserved.
```

```
Program : Function: WWWIisis
Syntax  : db=<dbname> [options]
Where   : dbname           ; data base name

Option 1: bool=<expression> ; search expression
Option 2: k1=<ifkey>         ; first inverted file key
          k2=<ifkey>         ; last inverted file key
Option 3: from=<mfnn>       ; first MFN
          to=<mfnn>         ; last MFN
Option 4: cgi=<fmt>         ; CGI format specification
Option 5: <update>         ; updating records
Others  : args             ; show next parameters
          hello            ; hello message
          menu=<n>         ; display group options
```

1.2. Versión de WWWISIS

La primera pantalla presenta la nota de copyright, la versión y los parámetros de instalación de la Biblioteca Interface CISIS con la que fue compilado WWWISIS y la versión de WWWISIS. Es importante chequear esta nota para identificar la versión actual de WWWISIS.

También presenta un menú de opciones de grupo de parámetros.

1.3. Sintaxis general de parámetro

WWWISIS procesa una base de datos según los parámetros pasados desde la línea de comandos y/o desde un fichero, especificado como argumento del parámetro *in*. En la siguiente línea de comando,

```
wwwisis db=cds bool=water$ pft=v70/
```

WWWISIS abrirá la base de datos *cds*, ejecutará la búsqueda booleana *water\$* y mostrará el campo 70 de los registros recuperados. El mismo resultado será obtenido emitiendo el siguiente comando:

```
wwwisis in=example.fil
```

donde *example.fil* es un archivo ASCII que contiene las siguientes líneas:

```
db=cds
bool=water$
pft=v70/
```

Una especificación de parámetro en un llamado de línea de comando debe encerrarse en comillas cuando su argumento contiene cualquier carácter que pueda ser interpretado como fin de una especificación de contenido de parámetro, como *space* o *'* o */*. Esto no se aplica cuando el parámetro es especificado en un archivo de entrada. Ejemplo:

```
wwwisis db=cds "bool=water and plant" from=1 "pft=v70|; |/"
```

Este ejemplo puede codificarse usando un archivo de entrada como sigue:

```
wwwisis in=param.in
```

donde *param.in* puede contener:

```
db=cds
bool=water and plant
from=1
pft=v70|; |/
```

Los nombres de parámetro de WWWISIS distinguen entre minúscula y mayúscula y todos deben colocarse en minúscula. Así, las palabras *BOOL* o *Bool* no serán reconocidas como parámetros, ya que la sintaxis correcta es *bool*.

El número máximo de parámetros en un llamado a WWWISIS es un parámetro constante de compilación, cuyo valor se muestra en la Pantalla 6 de la ayuda interna (código de retorno 10).

La posición real de los parámetros es irrelevante. Note sin embargo que, dependiendo del sistema operativo usado, la longitud máxima de una línea de comando es limitada; esta es

exactamente la razón por la que se recomienda el uso del parámetro *in* en lugar de la línea de comando.

Use el parámetro *args* para verificar los parámetros reales recibidos por WWWISIS.

Cuando los parámetros son generados "aal vuelo" usando el formato *cgi* o *cmd*, cada parámetro debe ser formateado en una nueva línea.

La descripción del contenido de cada parámetro aparece en la sección de referencia de parámetros de WWWISIS.

1.4. Parámetros obligatorios

La ejecución de WWWISIS recorre el conjunto de registros recuperados ejecutando las acciones especificadas por los parámetros.

En principio, es posible introducir cualquier combinación de parámetros WWWISIS, siempre que esta tenga sentido. Los únicos parámetros obligatorios son:

db - nombre de la base de datos;

pft - especificación de formato;

uno o más parámetros que recuperan datos de la base de datos, que pueden ser:

bool - expresión booleana, o

from to - rango de números secuenciales del archivo maestro, o

k1 k2 - rango de términos secuenciales del archivo invertido.

Cuando se especifica el parámetro *new*, no es obligatorio el uso de uno de los parámetros de recuperación anteriores.

1.5. Parámetro args y mensajes de error de WWWISIS

Con excepción de los errores de bajo nivel de la Biblioteca Interface CISIS (archivos no encontrados, error en archivos físicos, etc.), WWWISIS no emite ningún mensaje de error o advertencia en la salida. Cuando ocurre un error, WWWISIS retorna un código de salida diferente de cero. Es responsabilidad de la aplicación CGI dar tratamiento al código de salida. Este código se almacena en el campo de tag 1091 del registro log del WWWISIS, al que puede darse formato usando el parámetro *epilog*.

Durante la fase de desarrollo de una aplicación CGI WWWISIS, se recomienda el uso del parámetro posicional *args* que instruye a WWWISIS para listar los parámetros reales de ejecución. Esto es particularmente útil cuando los parámetros son generados dinámicamente por los parámetros *cgi* o *cmd*. Ejemplo:

```
wwwisis args cgi=@genparm.cgi
```

Además, la palabra clave de formato *ALL* puede usarse como herramienta de desarrollo para vaciar el contenido del registro virtual de ambiente CGI, los registros recuperados y el registro log. Ejemplo:

```
wwwisis db=cds cgi=ALL pft=, to=1
```

1.6. Grupos de parámetros

Los parámetros que dirigen la ejecución de WWWISIS se dividen en grupos diferentes. Estos grupos corresponden a las diferentes formas en que WWWISIS opera con la base de datos de entrada. Además, existe un grupo de parámetros generales, que pueden aplicarse en cualquier caso.

WWWISIS opera sobre uno o más registros del archivo maestro o términos del archivo invertido de la base de datos de entrada. La operación va recorriendo un registro o un término a cada momento. El conjunto de registros del archivo maestro puede obtenerse definiendo un rango de números del archivo maestro o como resultado de una expresión booleana o como una combinación de ambos métodos. El conjunto de términos del archivo invertido se obtiene definiendo un rango alfabético de términos. En el último caso es posible recorrer los postings de los términos.

Dependiendo de la forma en que se obtiene el conjunto de registros de entrada, existen diferentes grupos de parámetros que pueden usarse. Cada grupo corresponde a una pantalla de ayuda interna. Cada pantalla de ayuda tiene un número, según la lista siguiente:

- 8,9 - parámetros generales, pueden usarse con cualquier método de recuperación
- 1 - parámetros relacionados con expresión booleana
- 2 - parámetros relacionados con rango alfabético de términos del archivo invertido
- 3 - parámetros relacionados con rango secuencial de números de registros del archivo maestro
- 4 - parámetros relacionados con cgi y registro virtual de ambiente
- 5 - parámetros relacionados con la actualización de bases de datos

Además, están disponibles las siguientes pantallas:

- 0 - menú de grupos de parámetros
- 6 - códigos de retorno
- 7 - tags de campos de la base de datos log

2. Parámetros generales

Pantalla 8,9:

```
Options : proc=<fmt_proc>           ; CISIS field update format specification
          prolog=<fmt>              ; prologue format specification
          pft=<fmt>                 ; display format specification
          pfttag=<tag_field>        ; tag with display format specification
          pftgen=<fmt>              ; generating display format specification
          epilog=<fmt>              ; epilogue format specification
          gizmo=<dbname>[,<taglist>] ; gizmo data base name
          decod=<dbname>            ; decode data base name
          cipar=<cipar_file>        ; CISIS parameter file
          reverse                    ; display on reverse sequence
          step=<n>                   ; display every n
          count=<n>                  ; display n
          user=<user_code>           ; user identification
          pass=<user_password>      ; user password
          log=<dbname>              ; log data base name

Options : stop=<fmt>                ; semaphore format specification
          uctab=<file>               ; upper case character translation table
          actab=<file>               ; alphabetic character table
          in=<file>                  ; parameters input file name
          out=<file>                 ; output file name
          outapp=<file>              ; append output file name
          hlbool=<fst_spec>          ; fst for search highlight
          hltags=<fst_spec>          ; fst for tagged text highlight
          hltext=<fst_spec>          ; fst for text highlight
          maxlk=<n>                  ; maximum links (fst)
          create                     ; create data base
          createmst                   ; create master file
          createinv                   ; create inverted file
          callflag=<fmt>              ; customize exit code format specification
          next=<fmt>                 ; recursive call format specification
          quitpft=<fmt>              ; quit pft format specification
```

Los parámetros generales pueden ser usados con cualquier método para recuperar datos de la base de datos (búsqueda booleana, rango de mfn's o rango de claves invertidas). Incluyen el parámetro del nombre de archivo de los parámetros de entrada (*in*), los parámetros de nombre de archivo de datos de salida (*out*, *outapp*), los parámetros de edición de campos de datos (*proc* y *gizmo*), los parámetros de formato del registro virtual de ambiente CGI (*stop*, *prolog*), los parámetros de formato del registro recuperado (*pft*, *pfttag* y *pftgen*), selección del subconjunto de registros a procesar (*count*, *step*, *reverse* y *quitpft*), los parámetros relacionados con el control de proceso y la configuración (*cipar*, *files*, *user*, *pass*), los parámetros relacionados con el registro log del WWWISIS (*log*, *epilog*, *callflag* y *next*), manipulación del conjunto de caracteres (*uctab*, *actab*), parámetros de destaque del texto de salida (*hlbool*, *hltags* y *hltext*), parámetro de limitación del número de ligas (*maxlk*), inicialización de archivos de bases de datos (*create*, *createmst* y *createinv*).

Cuando se usan, el contenido de los parámetros de selección se agrega a los registros que están siendo procesados, por ejemplo:

1081 - parámetro *reverse*, si está presente contiene la palabra *reverse*

1082 - parámetro *step*

1083 - parámetro count

El contenido de los parámetros *user* y *pass* también se agrega al registro en los campos 1041 y 1042 respectivamente, lo que posibilita chequear la identificación y clave de entrada del usuario contra una base de datos existente para cada registro procesado así como nombrar los recursos temporales específicos durante una ejecución de WWWISIS, tales como el nombre de la base de datos de actualizaciones pendientes.

3. Parámetros relacionados con expresiones booleanas

Pantalla 1:

```
Option 1: bool=<expression>      ; search expression
          h1=<hit_number>         ; first hit
          h2=<hit_number>         ; last hit
          tpo=<tag>                ; current hit position tag      (1001)
          ttt=<tag>                ; total records tag          (1002)
          trp=<tag>                ; reverse current hit position tag (1003)
          ttl=<tag>                ; first hit tag              (1031)
          tt2=<tag>                ; last hit tag              (1032)
          tpm=<tag>                ; previous hit MFN tag      (1011)
          tbo=<tag>                ; search expression tag    (1021)
          tui=<tag>                ; user identification tag   (1041)
          tpa=<tag>                ; password tag            (1042)
          tdb=<tag>                ; data base name tag      (1051)
```

La búsqueda de registros usando expresiones booleanas es uno de los métodos que WWWISIS ofrece para recuperar información de bases de datos ISIS. La expresión booleana acepta cualquier operador de búsqueda ISIS (*, +, ^, (G), (F), ., \$). Una extensión de la interface Cisis permite también el uso de las palabras "and", "or", "and not". La siguiente expresión:

```
water * ( plant + plants) ^ transpiration
```

puede también escribirse como:

```
water and ( plant or plants) and not transpiration
```

(Note que el uso de palabras en inglés para representar los operadores booleanos en las expresiones de búsqueda es muy útil no sólo para facilitar las cosas a los usuarios si no también para evitar confusión con símbolos ya usados en el protocolo http, como el símbolo + para representar un espacio.)

Es posible mostrar sólo un conjunto de los registros recuperados usando los parámetros *h1* y *h2*. Estos son parámetros importantes en un ambiente cliente servidor para establecer un límite en la cantidad de datos transferidos de la base de datos al cliente. Si *h1* y *h2* no son especificados, WWWISIS tratará de formatear todos los registros recuperados. Los parámetros generales *count*, *from*, *to*, *step* y *reverse* pueden también aplicarse al límite del conjunto de registros recuperados a mostrar.

A no ser que se especifique el parámetro *reverse*, los registros serán mostrados en orden ascendente de los Mfns.

WWWISIS agrega a cada registro recuperado los siguientes campos de datos de procesamiento de información, por ejemplo:

- 1001 - posición del registro actual en modo no reverso
- 1002 - total de registros recuperados
- 1003 - posición del registro actual en modo reverso
- 1011 - Mfn del registro anterior
- 1031 - primer registro mostrado, o sea, el valor del parámetro *h1*
- 1032 - último registro mostrado, o sea., el valor del parámetro *h2*
- 1021 - the boolean expression

Si los registros de la base de datos ya tienen los campos anteriores, es posible cambiar los tags usando los parámetros *t*, como *tpo*, *tpp*, etc de WWWISIS.

Con los datos anteriores agregados a cada registro, es posible dar formato durante la salida no sólo a los campos de datos del registro como tal, si no también a los datos relacionados con el proceso de búsqueda. Además, esta facilidad permite transferir o ampliar estos datos en nuevas páginas HTML, a través de los elementos ocultos de entrada de un FORM.

El siguiente formato ejemplifica el uso de los campos de datos relacionados con el proceso de búsqueda:

```
.....
if val(v1001) = 1 then
  '<table>'/
  '<tr><td>Data Base:</td><td>', v1051, '</td></tr>', /
  '<tr><td>Search expression:</td><td>', v1021, '</td></tr>', /
  '<tr><td>Total Hits:</td><td>', v1002, '</td></tr>', /
  '<tr><td>Displaying:</td><td>', v1031, ' to ',
v1032, '</td></tr>', /
  '</table>',
fi,
.....
```

Se recomienda mucho chequear el código de retorno relacionado con la ejecución de una expresión booleana. El código de retorno se almacena en el campo de tag 1091 del registro log al final del proceso. El código de retorno puede ser chequeado por el script o a través del parámetro *epilog*.

4. Parámetros relacionados con rango alfabético de términos

Pantalla 2:

```
Option 2: k1=<ifkey>           ; first inverted file key
          k2=<ifkey>           ; last inverted file key
          posting[=<qtt>]      ; posting list
          notermtail          ; no summary record generation
          nomaster             ; no master file
          tck=<tag>            ; current key tag (1)
          tpt=<tag>            ; current key total postings tag (2)
          tps=<tag>            ; current posting structure tag (3)
          tpo=<tag>            ; current key position (1001)
          tt1=<tag>            ; first key tag (1031)
          tt2=<tag>            ; last key tag (1032)
          tui=<tag>            ; user identification tag (1041)
          tpa=<tag>            ; password tag (1042)
          tdb=<tag>            ; data base name tag (1051)
```

La búsqueda de un rango de términos del archivo invertido es uno de los métodos que WWWISIS ofrece para recuperar datos de bases de datos ISIS. Cada término y el correspondiente número de postings son almacenados respectivamente en los campos de datos 1 y 2 de un registro ISIS, que se mantiene sólo en memoria. Este registro siempre tiene Mfn igual a 1.

El parámetro general *count* puede también ser aplicado para limitar el conjunto de términos a mostrar. Este parámetro es importante porque proporciona un control más preciso sobre el número de términos a recuperar, además de los parámetros *k1* y *k2*. Note que no es necesario especificar *k1* y *k2*. Si se especifica sólo *k1*, se recuperan todos los términos desde *k1* hasta el final del archivo invertido, a no ser que *count* sea especificado. Si se especifica sólo *k2*, se recuperan todos los términos desde el inicio del diccionario del archivo invertido hasta *k2*, a no ser que *count* sea especificado. Es posible aplicar los parámetros *step* y *reverse* al conjunto de términos invertidos recuperados.

Cuando se recupera el último término, **WWWISIS crea una instancia adicional del registro en memoria - el registro resumen**, que contiene los siguientes campos de datos:

1002 - total de términos mostrados
1031 - primer término recuperado, más próximo a *k1*
1032 - último término recuperado, más próximo a *k2*

Para reconocer este registro, el formato puede verificar la presencia del campo de datos 1032 o la ausencia del campo de datos 1.

A continuación se muestra un ejemplo de una especificación de formato de registros de términos del archivo invertido:

```

if val(v1001) = 1 then '# Key',c35,'Postings' / fi,
v1001,x1,v1,c37,v2/
if a(v1) then
  'first key =',v1031/
  'last key =',v1032/
  'total kyes=',v1002/
fi/

```

5. Parámetros relacionados a un rango de registros del archivo maestro

Pantalla 3:

Option 3: from=<mf>	; first MFN	
to=<mf>	; last MFN	
tpo=<tag>	; current MFN tag	(1001)
ttt=<tag>	; total records tag	(1002)
trp=<tag>	; reverse current MFN tag	(1003)
ttl=<tag>	; first MFN tag	(1031)
tt2=<tag>	; last MFN tag	(1032)
tpm=<tag>	; previous MFN tag	(1011)
tui=<tag>	; user identification tag	(1041)
tpa=<tag>	; password tag	(1042)
tdb=<tag>	; data base name tag	(1051)

La búsqueda de un rango secuencial de registros del archivo maestro es uno de los métodos que ofrece WWWISIS para recuperar datos de bases de datos ISIS. Esta opción es particularmente útil para mostrar un registro específico cuyo Mfn es conocido. Los parámetros generales *count*, *step* y *reverse* pueden también usarse para controlar el conjunto de registros recuperados.

Los siguientes campos relacionados con el rango de registros recuperados son agregados al registro con información acerca del proceso de recuperación:

- 1001 - posición del registro actual en modo no reverso
- 1002 - total de registros recuperados
- 1003 - posición del registro actual en modo reverso
- 1011 - MFN del registro anterior
- 1031 - primer MFN mostrado, o sea, valor del parámetro *from*
- 1032 - último MFN mostrado, o sea, valor del parámetro *to*

A continuación se muestra un ejemplo de una especificación de formato de registros recuperados a través de un rango secuencial de Mfns, usando campos de datos agregados automáticamente por WWWISIS:

```

if a(v1081) and val(v1001) = 1 or p(v1081) and val(v1003) = 1 then
  ' mfn 1001 1002 1003 1011 1031 1032 reverse'/
fi,
mfn(5),f(val(v1001),5,0),f(val(v1002),5,0),f(val(v1003),5,0),
  f(val(v1011),5,0),f(val(v1031),5,0),f(val(v1032),5,0),
if p(v1081) then ' yes' else ' no' fi/

```

6. Parámetros relacionados con el registro virtual de ambiente CGI

Pantalla 4:

```
Option 4: cgi=<fmt>           ; CGI format specification
          env=<var>=<tag>[, ...]; environment variables
          cmd=<fmt>           ; command format specification
          tcg=<tag>           ; CGI transfer tag             (100)
          pfxtag=<text_id>    ; tag prefix to transfer values
          cttype=<info>      ; CGI content type
          get=<query_string> ; CGI QUERY_STRING simulation
```

Estos son parámetros opcionales relacionados con el registro virtual de ambiente CGI que WWWISIS construye cuando comienza la ejecución.

Se puede tener acceso al registro virtual a través de los parámetros *cgi* o *cmd*, que contienen las especificaciones de formato para generar dinámicamente los parámetros de ejecución de WWWISIS. Los parámetros *cgi* y *cmd* son exclusivos. Cuando está presente el parámetro *cgi*, WWWISIS agrega al registro virtual de ambiente CGI los datos transferidos por el servidor WWW a través de los métodos GET o POST. En el método POST, cuando se hacen varios llamados al WWWISIS en un script, el procesamiento *cgi* es válido sólo para el primer llamado; para los próximos, no existen datos disponibles en la entrada estándar.

El comando *env* se usa para agregar variables de ambiente al registro virtual de ambiente CGI. Un ejemplo de una variable añadida comúnmente, en un ambiente UNIX, es la identificación de proceso, que proporciona un número aleatorio que puede usarse, por ejemplo, para generar nombres únicos de archivos temporales en un ambiente multiusuario.

El parámetro *cgi* funciona sólo dentro de un ambiente CGI. Le indica al WWWISIS que debe procesar los datos que el servidor WWW transfiere al ambiente CGI. Como se indicó anteriormente, estos datos se espera que estén organizados en pares de elementos de datos correspondientes a los atributos de cualquier elemento HTML FORM INPUT, el primero identificando el *nombre* y el segundo identificando su *valor*.

7. Parámetros relacionados con la actualización de una base de datos

Pantalla 5:

```
Option 5: read=<lock_id>      ; read record lock id
          write=<lock_id>     ; write record lock id
          new                  ; new record
          fst=<fst_spec>       ; fst for inverted update
          pend=<dbname>       ; pending update data base name
          reread               ; allow same lock_id read again
          unlock=<seconds>    ; elapse seconds for unlock
          tul=<tag>            ; lock identification tag (1101)
          tus=<tag>            ; read/write lock status tag (1102)
          tum=<tag>            ; MFN tag (1103)

          read/write lock status possible values:
          0 - successful          4 - Record not locked
          1 - Data Entry Lock denied 5 - Lock id does not match
          2 - Record Lock denied   6 - Inverted Update denied
          3 - Record Update denied  7 - Lock expiration time not reached
```

Los parámetros de actualización permiten la creación de nuevos registros en la base de datos así como la modificación de los registros existentes.

Se agregan nuevos registros al final del archivo maestro y reciben el próximo número secuencial de archivo maestro (Mfn); así, si el último Mfn es 155, un nuevo registro tendrá un Mfn igual a 156. Cuando se usa el parámetro *new*, el uso de otros parámetros de recuperación es opcional, como en el siguiente ejemplo:

```
wwwisis db=cds new "proc='a10{campo 10{'" pft=mfn,x1,v10/
```

que crea un nuevo registro con el valor *campo 10* en el campo 10. Pero, cuando se usa el parámetro *new* y se recupera un conjunto de registros en el mismo llamado WWWISIS, se crea un registro nuevo para cada registro recuperado, por ejemplo:

```
wwwisis db=cds new from=1 to=10 pft=v70/
```

copiará los 10 primeros registros al final de la base de datos. Pueden usarse los parámetros *proc* y *gizmo* para modificar el contenido de los registros.

Las modificaciones a los registros existentes se realizan a través de los parámetros *read* y *write* que esperan una id de bloqueo (lock id) como argumento. El parámetro *read* trata de bloquear cada uno de los registros recuperados a modificar y si lo consigue, los registros son bloqueados y la id de bloqueo se coloca en el campo de datos 1101 (este tag puede cambiarse si fuera necesario, usando el parámetro *tul*). El parámetro *write* pide a WWWISIS la actualización de un registro recuperado si está bloqueado en ese momento y la id de bloqueo especificada en el parámetro *write* coincide con la id de bloqueo almacenada en el registro. Este mecanismo de actualización permite la actualización de registros en el mismo llamado WWWISIS o en un llamado diferente, lo cual es muy útil en un ambiente cliente-servidor sin estados y multiusuario como el WWW.

En el ejemplo siguiente, se recuperan los diez primeros registros de la base de datos CDS, se borra el campo 24 en cada registro y los registros se reescriben en el mismo llamado a WWWISIS;

```
wwwisis db=cds from=1 count=10 proc='d24' read=EME write=EME  
pft=v24/
```

El mismo ejemplo, puede ser ejecutado en dos pasos. En el primer paso, se llama a WWWISIS para recuperar los 10 primeros registros y asignarle una identificación de bloqueo a cada uno:

```
wwwisis db=cds from=1 count=10 read=EME pft=@updform.pft/
```

donde *updform.pft* puede contener una especificación de forma HTML a ser mostrada por el visualizador.

En el segundo paso, se llama a WWWISIS para escribir los mismos 10 registros recuperados con el contenido generado por el parámetro *proc* :

```
wwwisis db=cds proc=@updat.prc from=1 count=10 write=EME  
pft=@report.pft/
```

donde *updat.prc* contiene un procedimiento de actualización y *report.pft* contiene un reporte del proceso de actualización, tomando en cuenta el código de retorno del proceso de actualización (campo de tag 1102).

A veces, puede llamarse el mismo script CGI más de una vez, como por ejemplo cuando el usuario clicla el botón para recargar en la ventana del visualizador. Note que luego de que un registro es leído y bloqueado, no es posible leerlo nuevamente a no ser que se especifique el parámetro *reread*, indicando que WWWISIS puede leer de nuevo un registro bloqueado, asumiendo que la id de bloqueo coincida. Por ejemplo:

```
wwwisis db=cds proc=@updat.prc from=1 count=10 write=EME reread  
pft=@report.pft/
```

Los parámetros de actualización del WWWISIS están concebidos para operar siempre en un ambiente multi-usuario así como para facilitar el proceso de actualización en un contexto cliente-servidor sin estados. El código de retorno de la operación de actualización se almacena en el campo de tag. Se recomienda siempre chequear el código de retorno de actualización. Note que este código de retorno es diferente del código de salida de la ejecución del WWWISIS que es almacenado en el campo de datos 1091.

Cuando WWWISIS no puede escribir un registro en la base de datos, puede escribirlo en una base de datos alternativa especificada en el parámetro *pend*. La(s) base(s) de datos de actualización pendiente(s) debe(n) ser procesada(s) posteriormente. En el siguiente ejemplo, se asignan las bases de datos pendientes *data bases are assigned* al nombre de usuario:

```
wwwisis cgi=@update.cgi
```

donde el archivo *update.cgi* puede contener:

```
`db=cds`/  
`proc=@update.prc`/  
`bool=', (if v100^n = `bool` then v100^v fi),/  
`pend=', (if v100^n = `user` then v100^v fi),/  
`pft=@rec.pft`/  
'read=x`/  
'write=x`/
```

Como en un ambiente cliente-servidor sin estados es muy común que los registros permanezcan bloqueados, WWWISIS proporciona el parámetro *unlock* para desbloquear registros que permanecen bloqueados luego de transcurrido un período de tiempo determinado, independientemente de la identificación de bloqueo. Sólo son desbloqueados los registros bloqueados por WWWISIS. El siguiente llamado desbloquea registros que permanecen bloqueados durante dos o más horas, o 7,200 segundos o más:

```
wwwisis db=cds from=1 to=10 unlock=7200 pft=mn/
```

La actualización del archivo invertido se realiza cuando el parámetro *fst* es especificado con la correspondiente tabla de selección de campos.

8. Tags de campos del registro log

Pantalla 7:

Log fields:

cgi (1061)	step (1082)	maxlk (1121)	createinv (1244)
tcg (1062)	count (1083)	stop (1201)	hltags (1251)
env (1063)	posting (1084)	quit pft (1202)	hlbool (1251)
pfxtag (1064)	post qtt (1085)	uctab (1211)	hltext (1252)
pft (1071)	exit code(1091)	actab (1212)	freq (1271)
pfttag (1072)	read (1111)	call flag (1221)	freqv (1272)
pftgen (1072)	write (1112)	next (1222)	freqs (1273)
proc (1073)	fst (1113)	format error (1231)	
gizmo (1074)	unlock (1114)	fmt err place(1232)	
decod (1075)	tul (1115)	proc err plac(1233)	
files (1076)	tus (1116)	srch err code(1234)	
cipar (1077)	new (1117)	srch err plac(1235)	
epilog (1078)	tum (1118)	notermtail (1241)	
prolog (1079)	pend (1119)	nomaster (1242)	
reverse(1081)	reread (1120)	createmst (1243)	program version(1301)

Para cada llamado, WWWISIS crea un registro log justo antes de terminada su ejecución. Este registro contiene campos de datos que almacenan el contenido de los parámetros de ejecución así como el resultado del proceso.

El registro log puede formatearse usando el parámetro *epilog* y/o almacenarse en una base de datos usando el parámetro *log* . Por ejemplo, el siguiente llamado WWWISIS:

```
wwwisis db=cds "bool=water and plant" pft=@cds.pft log=xlog epilog=ALL
```

creará la base de datos *xlog* con el registro Mfn 1 conteniendo los siguientes campos de datos formateados según el parámetro *epilog*:

```
mfn=1
1071 «MFN(4), ' - ',MDL,V12,V24,,((|V76^Z|: |,V76^*|) |),V70+|;
|,V25,V26,V30,|
(|V44|) |,V50,/,|/| |V71/|/| |V72/|/| |V74/"KEYWORDS: ",V69(10,10)/##»<BR>
1078 «ALL»<BR>
1082 «1»<BR>
1083 «2147483647»<BR>
1115 «1101»<BR>
1116 «1102»<BR>
1118 «1103»<BR>
1091 «0»<BR>
1002 «2»<BR>
1021 «water and plant»<BR>
1051 «cds»<BR>
```

Use el parámetro *epilog* con la palabra clave *ALL* para conocer los campos de datos del registro log presentes al final de una corrida de WWWISIS.

9. Códigos de retorno WWWISIS

Pantalla 6:

Return codes:

0 - ok	18 - actab	36 - proc field update
1 - fatal	19 - format	37 - pl01
2 - field update	20 - last MFN	38 - callflag format
3 - search syntax	21 - frequency alloc	39 - input file
4 - search run	22 - epilog	40 - FST file
5 - format syntax	23 - record update	41 - pft_gen file
6 - format run	24 - new record	42 - stop pft file
7 - 0 records retrieved	25 - update	43 - cgi/cmd proc file
8 - format is mandatory	26 - unlock	44 - format open
9 - CIGI error	27 - hits	45 - search log
10 - MAX_ARGV=80	28 - gizmo tag	46 - HL parameter missing
11 - data base name	29 - output file	47 - HL fst generation
12 - pftgen	30 - output buffer size	48 - HL text change
13 - pending not saved	31 - cgi proc	49 - hit record
14 - couldn't start	32 - cgi format	50 - no parameters found
15 - gizmo quantity	33 - stop format	51 - next format
16 - cipar	34 - prolog format	52 - quit pft
17 - uctab	35 - proc format	

La aplicación WWWISIS no emite ningún mensaje de error, con excepción de los mensajes de error de bajo nivel generados directamente por la Biblioteca Interface Cisis (que incluye, por ejemplo, número máximo de ficheros abiertos excedido, memoria RAM agotada, archivos de bases de datos no encontrados, etc.). Los códigos de retorno de WWWISIS pueden ser procesados a través del formato *epilog* o directamente por el script después de cada llamado.

Es muy recomendable incluir el parámetro *epilog* en cada llamado a WWWISIS para controlar su código de salida, que es almacenado en el campo de tag 1091 del registro *log*. A continuación se muestra un ejemplo del formato epílogo:

```
,If v1091 <> 0 then v1091 <> 7 then
  ,"<br> Error: código de retorno="v1091"<br><blockquote>"
    If v1091 = '1' then 'Error fatal' else
    If v1091 = '2' then '<b>proc<b> invalida!' else
    If v1091 = '3' then '<b>pft<b> invalido!' else
      ...
    ,fi
  fi,
fi,
,'</blockquote>'
,fi,
```

10. Parámetros reservados

WWWISIS tiene varios parámetros adicionales reservados exclusivamente para uso interno de BIREME. Se listan a continuación sólo para información:

```
cipchange=<find_char><replace_char>
decod=<dbname>
gizad=<dbname>[, <taglist>]
hits
p101=<proc_101>
searchlog=<dbname>
searchnumber=<number>
searchtagid=<tag>
slogfu=<field_update>
tmpf=<file>
tw2mh=<1/0>
```

III. Manual de referencia de parámetros de WWWISIS

?

Desde la versión 3.0 de WWWISIS, el ? ha sido sustituido por *args* .

actab=<file> ; ascii characters table

El <file> representa un nombre de archivo que contiene una tabla de caracteres alfabéticos de ISIS, o sea, una lista de códigos de caracteres ascii a ser interpretados como parte de una palabra para la extracción de términos usando las técnicas de inversión 4 y 8. Todos los caracteres no incluidos en la tabla son interpretados como separadores de palabras. El parámetro *actab* es activado por el parámetro *fst* y usado cuando hay una extracción de palabra presente en la tabla de selección de campos.

Ejemplo:

```
wwwisis db=dbx actab=/isis/menu/isisac.tab new proc='a10{xxx{'  
fst=cds.fst
```

La tabla alfabética por defecto del WWWISIS coincide con la distribuida con MicroISIS.

Al definir una nueva tabla alfabética, use siempre tres dígitos para cada entrada de código de carácter ascii. Cada línea, exceptuando la última, debe contener 32 entradas.

Ejemplo:

```
065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080 081 082 083 084  
085 086 087 088 089 090 097 098 099 100 101 102  
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122  
128 129 130 131 132 133 134 135 136 137 138 139  
140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 160 161 162 163 164  
165
```

args ; display parameters

El parámetro *args* vacía el contenido de los parámetros WWWISIS que le siguen en la línea de comando. El parámetro *args* puede ser usado en cualquier posición de la línea de comando.

Se recomienda el uso del parámetro *args* para detección de errores, particularmente cuando los parámetros de ejecución son generados dinámicamente a través de los parámetros *cgi* o *cmd*.

Ejemplos:

```
wwwisis args cgi=@param.cgi
```

```
wwwisis db=cds args bool=water pft=@cds.pft
```

bool=<expression> ; search expression

La <expression> representa o una especificación de búsqueda booleana ISIS in-line o un nombre de archivo precedido por @ , conteniendo una especificación de búsqueda booleana ISIS.

El parámetro bool permite la recuperación de registros con base en una expresión booleana que combine términos del archivo invertido. De acuerdo con la estructura del archivo invertido ISIS, cada ocurrencia de un término del archivo invertido es almacenada en el archivo invertido según una estructura de datos llamada posting, que incluye cuatro elementos: Mfn (número de archivo maestro donde ocurre el término), Id (número de identificación del término, como aparece codificado en la FST), Occ (Ocurrencia del término) y Pos (posición del término).

Dados dos términos, A y B, el Lenguaje de Búsqueda de ISIS acepta los siguientes operadores Booleanos listados de menor a mayor prioridad en la evaluación de la expresión:

Operación	Operador	Ejemplo	Elementos de Posting (Mfn, Id, Occ, Pos) considerados en la operación Booleana
Or	+	A+B	Mfn: el resultado es la suma de todos los postings de A y B
And	*	A*B	Mfn: el resultado está formado por los postings de A y B que tienen el mismo Mfn
And Not	^	A^B	Mfn: el resultado está formado por los postings de A con Mfn no presente en los postings de B
Group	(G)	A(G)B	Mfn e Id: el resultado está formado por los postings de A y B que tienen el mismo Mfn e Id
Field	(F)	A(F)B	Mfn, Id y Occ: el resultado está formado por los postings de A y B que tienen el mismo Mfn, Id y Occ
Adjacency	.	A . B	Mfn, Id, Occ y Pos: el resultado está formado por los postings de A y B que tienen el mismo Mfn, Id, Occ y una diferencia entre la Pos correspondiente no mayor que el número de puntos
Adjacency	\$	A \$ B	Mfn, Id, Occ y Pos: el resultado está formado por los postings de A y B postings que tienen el mismo Mfn, Id, Occ y una diferencia entre la Pos correspondiente exactamente igual al número de signos de dólar

La Interface Cisis permite el uso de las palabras **AND** o **and** , **OR** o **or** , **AND NOT** o **and not** en lugar de *, + , ^ respectivamente. El carácter \$ puede usarse para buscar una raíz, como por ejemplo *plant\$*. Pueden usarse paréntesis para definir el orden de evaluación de la expresión. Ver la documentación de CDS/ISIS para mayor información.

Ejemplos:

```
bool=water and (plant or plants) and not transpiration
```

```
bool=water * (plant + plants) ^ transpiration
```

```
bool=@boolean.exp
```

callflag=<fmt> ; change exit code format

El <fmt> representa o una especificación de formato in-line o un nombre de fichero precedido por el carácter @ conteniendo una especificación de formato. La salida del formato debe ser un número que represente el código de salida. Generalmente se usa para adaptar los códigos de salida de WWWISIS.

Ejemplo:

```
wwwisis db=cds "bool=water ++" pft=@cds.pft callflag=@exit.flg epilog=@cds.epi
```

donde el *exit.flg* contiene:

```
if val(v1091) = 7 then '0' else v1091 fi
```

y *cds.epi* contiene:

```
,select s(v1091)
  case '0': ,
  case '3': 'Error: search syntax!'/,
  case '4': 'Error: search run-time exception!'/,
  case '5': 'Error: format syntax!'/,
  case '6': 'Error: format run-time exception!'/,
  elsecase "Error code "v1091,
,endsel,
```

cgi=<fmt> ; CGI format parameters

El <fmt> representa o una especificación de formato in-line o un nombre de fichero precedido por el carácter @ conteniendo una especificación de formato. La salida del formato es interpretada como nuevos parámetros para WWWISIS. Cada nuevo parámetro debe ser formateado en una nueva línea. La especificación de formato actúa sobre el registro virtual ISIS de ambiente CGI que WWWISIS crea cuando comienza.

Ejemplos:

```
wwwisis cgi=@filepar.cgi
```

```
wwwisis "cgi='db=cds'/', 'from=1'/', 'to=3'/', 'pft=mfn/'"
```

```
wwwisis db=cds "cgi='from=1'/', 'to=3'/', 'pft=mfn/'"
```

cipar=<cipar_file> ; CISIS parameter file

El <cipar_file> representa el nombre de un archivo de parámetros de Cisis.

Un archivo de parámetros de CISIS es un archivo ASCII usado en WWWISIS para redefinir las rutas de acceso y nombres de los archivos. Cada línea del fichero de parámetros

representa un nombre de entrada lógico seguido por el signo = y el nombre de la salida física.

Ejemplo:

```
CDS.*=c:\isis\data\cds\cds.*
```

traducirá todos los nombres lógicos de archivos *CDS* en nombres físicos de archivos *c:\isis\data\cds*.

El contenido del fichero de parámetros CISIS reconoce mayúsculas y minúsculas. Por tanto, en el ejemplo, *CDS.mst* se traducirá como *c:\isis\data\cds*, mientras que *cds.mst* no será.

Las entradas lógicas pueden usarse en los siguientes parámetros WWWISIS: *bool*, *cgi*, *cmd*, *db*, *epilog*, *fst*, *gizmo*, *log*, *pft* y *proc*.

Ejemplo:

El fichero de parámetros CISIS puede usarse para permitir acceso a los archivos físicos maestro e invertido con nombres diferentes:

```
wwwisis cipar=author.cip db=cds bool=magalhaes$ pft=mfn/
```

donde el archivo *author.cip* contiene:

```
cds.mst=/bases/cds/cds.mst
cds.xrf=/bases/cds/cds.xrf
cds.cnt=/bases/cds/author.cnt
cds.cnt=/bases/cds/author.cnt
cds.n01=/bases/cds/author.n01
cds.n02=/bases/cds/author.n02
cds.l01=/bases/cds/author.l01
cds.l02=/bases/cds/author.l02
cds.ifp=/bases/cds/author.ifp
```

cmd=<fmt> ; environment format parameters

El <fmt> representa o una especificación de formato in-line o un nombre de fichero precedido por el carácter @ conteniendo una especificación de formato. La salida del formato es interpretada como nuevos parámetros al WWWISIS. Cada nuevo parámetro debe ser formateado en una nueva línea. La especificación de formato actúa sobre el registro virtual ISIS de ambiente CGI que WWWISIS crea cuando comienza.

Los parámetros *cmd* y *cgi* son excluyentes. Por tanto, cuando está presente el parámetro *cmd*, WWWISIS no agregará automáticamente el campo de tag 100 al registro virtual de ambiente CGI.

Cuando se especifica el parámetro *cmd*, la única forma de agregar datos al registro virtual es usando el parámetro *env* para agregar el valor de las variables de ambiente. También es

posible formatear los registros desde diferentes bases de datos físicas usando los comandos de formato *ref* (*reference*) y *l* (*look-up*).

Ejemplos:

```
wwwisis cmd=@filepar.cgi
```

```
wwwisis "cmd='db=cds' /, 'from=1' /, 'to=3' /, 'pft=mfn/' "
```

```
wwwisis db=cds "cmd='from=1' /, 'to=3' /, 'pft=mfn/' "
```

count=<n> ; display n

La <n> representa el número máximo de registros recuperados a procesar por WWWISIS. Si el número de registros recuperados es menor que <n>, todos los registros serán procesados.

Ejemplos:

```
wwwisis db=cds bool=a$ count=10 pft=mfn/
```

```
wwwisis db=cds k1=A k2=AZZZZ count=10 pft=v1/
```

create ; create data base
createinv ; create inverted files
createmst ; create master files

Estos parámetros se usan para crear archivos maestros y/o invertidos correspondientes al nombre de la base de datos especificado en el parámetro *db*. Note que si el fichero maestro y/o invertido ya existen, serán inicializados.

Ejemplo:

```
wwwisis db=y create pft=,  
wwwisis db=z createmst pft=,  
wwwisis db=z createinv pft=,
```

ctttype=<info> ; CGI content type

El <info> representa el tipo de contenido del texto de salida de WWWISIS. Cuando está presente el parámetro *ctttype*, WWWISIS produce automáticamente como salida dos líneas correspondientes al encabezamiento del documento http. La segunda línea es siempre una línea en blanco.

Ejemplo:

```
wwwisis ctttype=text/plain menu=6
```

db= <dbname> ; data base name

El <dbname> representa nombre de base de datos lógico o físico. Si <dbname> es un nombre lógico, hay que especificar el fichero de parámetros Cisis.

El nombre de la base de datos es un parámetro obligatorio en cada llamado al WWWISIS.

Si el nombre de la base de datos y/o la ruta de acceso son incorrectos, CISIS emite el siguiente mensaje:

```
dbxopen: 'cds.xrf' errno=2
fatal: dbxopen/open
```

Ejemplos:

```
wwwisis db=c:\bases\cds\cds bool=water$ pft=@cds.pft
```

```
wwwisis cipar=files.par db=CDS bool=water$ pft=@cds.pft
```

En este caso, una de las líneas de *files.par* es:

```
CDS.*= c:\bases\cds\cds.*
```

env=<var>=<tag>[, ...] ; environment variables

El <var> representa un nombre de variable de ambiente y <tag> un tag de campo de datos en el registro virtual de ambiente CGI, donde será agregado el valor de la variable de ambiente. Este registro virtual está disponible cuando WWWISIS comienza y puede ser formateado a través de *cgi* o *cmd*. Es posible agregar una o más variables de ambiente.

Exjemplo:

```
export SYSTEM=test
export PROCESS=$$
wwwisis env=PROCESS=201,SYSTEM=202 cgi=@genpar.cgi
```

donde *genpar.cgi* podría contener:

```
'db=',if v202 = 'test' then 'test' else 'cds' fi /
'bool=water$'/
'pft=@cds.pft'/
'log=L', v201/
```

epilog=<fmt> ; epilogue format

El <fmt> representa o una especificación de formato in-line o un nombre de fichero precedido por el carácter @ conteniendo una especificación de formato. La salida del formato es enviada a la salida estándar. La especificación de formato es aplicada al registro log virtual que WWWISIS crea cuando termina su ejecución.

Ejemplo:

```
wwwisis db=cds bool=water$ pft=mfn/ epilog=@rc.pft
```

onde *rc.pft* pudiera contener:

```
if val(v1091) = 7 then ,
    ' No records retrieved.' /,
else,
    if val(v1091) = 3 then ,
        'Search syntax error' /,
    else,
        if val(v1091) = 4 then
            'Search execution error' /
        fi,
    fi,
fi,
```

files=<n> ; maximum open files

La <n> representa el número máximo de ficheros abiertos MS/DOS necesarios en un llamado a WWWISIS. El valor por defecto de MS/DOS es 20. Como WWWISIS puede acceder varias bases de datos en una corrida, este parámetro permite la modificación del valor por defecto. El parámetro correspondiente en el CONFIG.SYS, FILES=<nfiles>, debe tener un valor asignado previamente, con <nfiles> mayor o igual que <n> .

Ejemplo:

```
wwwisis files=80 db=cds pft=@cads.pft bool=water$
```

El parámetro *files* se aplica sólo a la versión de WWWISIS para 16 bits.

from=<mf> ; first MFN

El <mf> representa el menor Número de Archivo Maestro (MFN) a procesar.

Ejemplos:

```
wwwisis db=cds from=10 count=3 pft=mf/
```

```
wwwisis db=cds from=10 count=3 pft=mf/ reverse
```

```
wwwisis db=cds from=10 to=13 pft=mf/
```

```
wwwisis db=cds from=10 to=13 pft=mf/ reverse
```

```
wwwisis db=cds bool=w$ from=5 to=7 pft=mf/
```

fst=<fst_spec> ; fst for inverted update

La <fst_spec> representa una especificación in-line de Tabla de Selección de Campos (FST) o un nombre de archivo, precedido por el carácter @, y conteniendo una Tabla de Selección de Campos.

El parámetro *fst* invierte nuevos registros creados con el parámetro *new* o registros existentes actualizados usando el parámetro *write*.

Se recomienda chequear el código de retorno de la operación de inversión. Los códigos de retorno de la actualización de registros se almacenan en el campo 1102 y pueden contener los siguientes valores:

- 0 - Exito
- 1 - Bloqueo para entrada de datos negado
- 2 - Bloqueo de registro negado
- 3 - Actualización de registro negada
- 4 - Registro no bloqueado
- 5 - Identificación de bloqueo no coincide
- 6 - Actualización de invertido negada
- 7 - No se llegó al tiempo de expiración del bloqueo

Ejemplo de la creación de un nuevo registro:

```
wwwisis in=newrec
```

donde el fichero *newrec* contiene:

```
db=cds
new
proc='a24{Magalhaes, Elisabeth{a24{A wwwisis advanced application{
fst=@cfs.fst
pft=@chkrc
```

Ejemplo de actualización de un registro existente:

```
wwwisis db=cds read=xxx proc='d*' from=156 fst=@cds.fst write=xxx  
pft=@chkrc
```

donde *chkrc* puede contener:

```
if val(v1102) <>6 then ' Inverted file NOT done! rc=', v1102/fi
```

get=<string> ;CGI QUERY_STRING simulation

El <string> es incorporado a la variable de ambiente QUERY_STRING. Si el parámetro *cgi* está presente, el contenido de QUERY_STRING se agrega al registro virtual de ambiente CGI que WWWISIS crea cuando comienza. El propósito del parámetro *get* es simular el ambiente CGI en un llamado de línea de comando al WWWISIS, de forma que pueda ser usado con fines de prueba o detección de errores. El parámetro *get* simula el método GET de http de transferencia de datos.

Ejemplo:

```
wwwisis "get=db=cds&bool=plants and water"  
"cgi=(v100^n,|=|v100^v/)" pft=@cds.pft
```

gizmo=<dbname>[,<taglist>] ; gizmo data base name

El <dbname> representa el nombre de un archivo maestro ISIS que contiene una tabla de cambios globales de gizmos y <taglist> representa una lista opcional de tags a cambiar. La base de datos de gizmos contiene un registro para cada gizmo; un gizmo es codificado en los campos de tag 1 y 2 y, opcionalmente, los campos de tags 11 y 21, de acuerdo con:

tag	Contenido
1	Valor de entrada a cambiar
2	Valor de salida si no se encuentra el valor de entrada
11	valor <i>asc</i> si los caracteres del campo 1 están representados por sus códigos ascii valor <i>hex</i> si los caracteres del campo 1 están representados por su código hexadecimal
21	Valor <i>asc</i> si los caracteres del campo 2 están representados por sus códigos ascii valor <i>hex</i> si los caracteres del campo 1 están representados por sus códigos hexadecimales

Ejemplo:

```
wwwisis db=cds gizmo=html bool=sedimentation pft=v24/
```

donde la bases de datos de gizmos *html* puede contener los siguientes registros:

```
mfn= 1
  1 «é»
  2 «&eacute;i»
```

```
mfn= 2
  1 «è»
  2 «&egrave;i»
```

h1=<hit_number> ; first hit

El <hit_number> representa el menor número de registro hallado en una búsqueda (hit) a procesar. Si el número de registros recuperados es menor que <hit_number> no se procesa ningún registro.

Ejemplo:

```
wwwisis db=cds bool=w$ h1=11 h2=20 pft=@cds.pft
```

h2=<hit_number> ; last hit

El <hit_number> representa el mayor número de registro hallado en una búsqueda (hit) a procesar, si existe.

Ejemplo:

```
wwwisis db=cds bool=w$ h1=11 h2=20 pft=@cds.pft
```

hello ; hello message

Cuando este parámetro está presente, WWWISIS produce como salida un mensaje de saludo. Este parámetro es generalmente usado con fines de prueba.

Ejemplo:

```
http://cgi-bin/wwwisis.exe/[hello]/
```

hlbool=<fst_spec> ; fst for search high light hltext=<fst_spec> ; fst for text high light

La <fst_spec> representa una especificación in-line de Tabla de Selección de Campos (FST) o un nombre de archivo, precedido por el carácter @, y conteniendo una Tabla de Selección de Campos. Estos parámetros se usan para identificar en los registros recuperados los términos de búsqueda introducidos como argumento del parámetro *bool*. Los términos de búsqueda identificados pueden ser resaltados en el texto usando elementos HTML.

La fst *hlbool* es aplicada a cada registro recuperado para generar ligas que coincidan con los términos del archivo invertido. Las ids de las ligas generadas deben coincidir con los tags de campos de datos reales de donde se extraen las ligas. Note que la ids de liga en las fsts usadas para generar archivos invertidos no necesariamente coinciden con los tags de campos de datos. Por ejemplo, la base de datos CDS puede invertirse usando la siguiente fst:

```
1 0 mp1, ( |AU= |V70 / )
2 4 mp1, V24, / , v69 /
3 2 mp1, V69 /
```

La fst *hlbool* debe usar los tags de campos de datos reales para identificar las ligas:

```
70 0 mp1, ( |AU= |V70 / )
24 4 mp1, V24, /
69 4 mp1, v69 /
69 2 mp1, V69 /
```

Esta fst genera una lista de ligas para cada registro recuperado. Estas ligas deben coincidir con los términos reales del archivo invertido. En la mayoría de los casos, la fst *hlbool* es igual o muy parecida a la fst usada para generar el archivo invertido, incluyendo los prefijos que pueden ser usados para calificar los términos invertidos.

WWWISIS analiza los términos de búsqueda y trata de identificar cada término de búsqueda en la lista de ligas producida por la fst *hlbool*. Cuando se encuentra un término de búsqueda en esta lista de ligas, WWWISIS trata de localizarlo en el campo de datos correspondiente. Cuando se localiza un término de búsqueda, WWWISIS agrega un prefijo y un sufijo según la especificación producida por la fst *hltext*.

La fst *hltext* se aplica a cada registro recuperado para generar ligas que contienen la especificación de las cadenas de prefijos y sufijos a agregar a los términos de búsqueda encontrados en los tags de campos de datos correspondientes de los registros recuperados. Los prefijos y sufijos son especificados en las ligas generadas a través de una estructura de subcampos, de acuerdo con la siguiente tabla:

Subcampo	Descripción
^1	prefijo
^2	sufijo
^3	identificador de subcampo de datos donde se localiza el término de búsqueda, si se aplica
^4	prefijo de término de búsqueda, no presente en los datos, si se aplica
^5	sufijo de término de búsqueda, no presente en los datos, si se aplica
^6	tabla de traducción de términos de búsqueda, si se aplica

La siguiente fst *hltext* permite mostrar los términos de búsqueda en azul:

```
70 0 ' ^1<FONT COLOR=BLUE>^2</FONT>^4AU= '
24 0 ' ^1<FONT COLOR=BLUE>^2</FONT> '
69 0 ' ^1<FONT COLOR=BLUE>^2</FONT> '
```

hltags=<fst_spec> ; fst for tags high light

La <fst_spec> representa una especificación in-line de Tabla de Selección de Campos (FST) o un nombre de archivo, precedido por el carácter @ , y conteniendo una Tabla de Selección de Campos. Estos parámetros se usan para identificar en los registros recuperados las cadenas generadas por la fst. Las cadenas identificadas pueden ser resaltados en el texto usando elementos HTML. Note que el parámetro *hltags* no requiere del parámetro *bool*, o sea, se aplica a todos los métodos de recuperar registros.

La fst *hltags* genera una lista de ligas para cada registro recuperado. WWWISIS trata de localizar cada cadena de liga en el campo de datos correspondiente. Cuando se localiza una cadena, WWWISIS agrega un prefijo y un sufijo de acuerdo con la especificación producida por la fst *hltext*.

Ejemplo:

```
wwwisis db=cds to=5 pft=v24/ "hltags=24 0 mpl,V24" "hltext=24 0 '^1<B>^2</B>'"
```

in=<input_file>

El <input_file> representa el nombre de un archivo que contiene los parámetros de ejecución de WWWISIS. Cada línea del archivo representa un parámetro. La técnica de usar un archivo *in* para parametrizar WWWISIS puede ser usada en todos los métodos de llamado al WWWISIS. Es particularmente apropiado cuando se pasa al ambiente CGI usando la variable de ambiente CGI PATH_INFO, la forma recomendada de llamar al WWWISIS en Windows NT IIS.

Ejemplos:

```
http://cgi-bin/wwwisis.exe/[in=goparm.in]/
```

```
wwwisis in=param.in
```

Donde *param.in* puede contener:

```
args
db=cds
bool=water$
pft=mfn/
```

El siguiente ejemplo muestra que el parámetro de archivo de entrada puede ser usado junto con parámetros colocados en la línea de comando y/o generados a través del archivo de parámetros *cgi*:

```
wwwisis db=cds in=param.in
```

Donde *param.in* contiene:

```
prolog=@cds.pro
cgi=@cds.cgi
epilog=@cds.epi
```

k1=<ifkey> ; first inverted file key

El <ifkey> representa el menor término del archivo invertido a recuperar. Si no está presente en el archivo invertido, se retorna el próximo término. Cada término de archivo invertido recuperado es almacenado en el campo de tag 1 de un registro de archivo maestro ISIS y el número correspondiente de postings en el campo de tag 2. Este registro se mantiene sólo en memoria y tiene siempre Mfn igual a 1. Si el parámetro *posting* está presente, WWWISIS recorre todos los postings de cada término y almacena los elementos del posting en el campo de tag 3.

Si se especifica el parámetro *notermtail*, WWWISIS crea una instancia adicional de registro después de procesado el último término - un registro resumen, que contiene los siguientes campos:

1002 - número total de términos mostrados
1031 - primer término recuperado, más próximo de k1
1032 - último término recuperado, más próximo de k2

Ejemplo:

```
wwwisis db=cds k1=w pft=@showks
```

donde *showks* contiene:

```
if val(v1001) = 1 then 'Postings Key' fi /,
if p(v1) then f(val(v2),5,0),x7,v1/,
else ,
  ' first key= ',v1031/
  ' last key= ',v1032/
  ' total keys= ',v1002/,
fi
```

Cuando no se especifica ningún argumento para el parámetro *k1*, WWWISIS comienza con el primer término del archivo invertido. Ejemplo:

```
wwwisis db=cds k1= step=100 pft=v1/
```

k2=<ifkey> ; last inverted file key

El <ifkey> representa el mayor término del archivo invertido a recuperar. Si no se encuentra en el archivo invertido, se considera el mayor término anterior como el término mayor. Cada término recuperado del archivo invertido es almacenado en el campo de tag 1 de un registro

de archivo maestro ISIS y el número correspondiente de postings en el campo de tag 2. Este registro se mantiene sólo en memoria y tiene siempre Mfn igual a 1. Si el parámetro *posting* está presente, WWWISIS recorre todos los postings de cada término y almacena los elementos del posting en el campo de tag 3.

Si se especifica el parámetro *notermtail*, WWWISIS crea una instancia adicional del registro después de procesado el último término, que contiene los siguientes campos:

- 1002 - número total de términos mostrados
- 1031 - primer término recuperado, más próximo de k1
- 1032 - último término recuperado, más próximo de k2

Ejemplo:

```
wwwisis db=cds k2=ZZZZZZZZZZZZZ pft=@lowhigh
```

donde *lowhigh* contiene:

```
if a(v1) then
  ' lowest key found was ',v1031/
  ' highest key found was ',v1032/
fi
```

log=<dbname> ; log data base name

El <dbname> representa el nombre de un archivo maestro ISIS donde WWWISIS escribirá el registro log, creado al final del proceso. El registro log puede contener los siguientes campos de datos:

cgi	(1061)
tcg	(1062)
env	(1063)
pfxtag	(1064)
pft	(1071)
pfttag	(1072)
pftgen	(1072)
proc	(1073)
gizmo	(1074)
decod	(1075)
files	(1076)
cipar	(1077)
epilog	(1078)
prolog	(1079)
reverse	(1081)
step	(1082)
count	(1083)
posting	(1084)
post qtt	(1085)
exit code	(1091)

read	(1111)
write	(1112)
fst	(1113)
unlock	(1114)
tul	(1115)
tus	(1116)
new	(1117)
tum	(1118)
pend	(1119)
reread	(1120)
maxlk	(1121)
stop	(1201)
quit pft	(1202)
uctab	(1211)
actab	(1212)
call flag	(1221)
next	(1222)
format error	(1231)
fmt err place	(1232)
proc err plac	(1233)
srch err code	(1234)
srch err plac	(1235)
notermtail	(1241)
nomaster	(1242)
createmst	(1243)
createinv	(1244)
hltags	(1251)
hlbool	(1251)
hltext	(1252)
program version	(1301)

Ejemplos:

```
wwwisis db=cds from=1 reverse pft=mf,',' log=xlog
```

```
wwwisis db=xlog from=1 pft=ALL
```

maxlk=<n> ; **maximum link terms (fst)**

El argumento <n> representa el número máximo de ligas que WWWISIS debe generar cuando aplica una especificación *fst* en un registro. El valor por defecto es 500 ligas por registro.

Ejemplo:

```
wwwisis db=text from=1 to=5 read=x "fst=1 4 mpl,v1/" write=x
pft=v1102/ maxlk=1000
```


menu=<n> ; specific parameters menu

La <n> representa el número del menú interno que WWWISIS debe mostrar.

Ejemplos:

```
wwwisis menu=6
```

new ; new record

El parámetro *new* crea un nuevo registro en la base de datos definida en el parámetro *db* . El Mfn del nuevo registro es el próximo número secuencial después del mayor Mfn de la base de datos. WWWISIS requiere el bloqueo de la entrada de datos para crear con éxito un nuevo registro. Use el parámetro *fst* para invertir el registro recién creado. El campo de información de tag 1102 contiene el código de retorno de la operación, que puede ser:

- 0 - Exito
- 1 - Bloqueo para entrada de datos negado
- 2 - Bloqueo de registro negado
- 3 - Actualización de registro negada
- 4 - Registro no bloqueado
- 5 - Identificación de bloqueo no coincide
- 6 - Actualización de invertido negada
- 7 - No se llegó al tiempo de expiración del bloqueo

Ejemplos:

```
wwwisis db=cds new pft=@checkrc
```

donde *checkrc* puede contener:

```
if val(v1102) <> 0 then 'Record not created: rc=',v1102,'!', fi
```

next=<fmt> ; wwwisis loop

El <fmt> representa una especificación de formato in-line o un nombre de archivo, precedido por el carácter @, y conteniendo una especificación de formato. Este formato es aplicado al registro LOG se comporta como el formato del parámetro *cgi* ya que su salida es interpretada como un llamado recursivo a WWWISIS.

Ejemplo:

```
wwwisis db=cds to=10 pft=mfn/  
"next='db=thes' /, 'to=10' /, 'pft=mfn/' /"
```

nomaster ; no master files

Este parámetro se usa para indicar que no existe archivo maestro para el nombre de base de datos especificado en el parámetro *db*. Se usa junto con los parámetros *k1* y/o *k2* para recuperar términos de archivos invertidos sin archivo maestro correspondiente.

Ejemplo:

```
wwwisis db=author k1=a nomaster pft=v1/
```

notermtail ; no term tail generation

Este parámetro se usa para deshabilitar la generación del registro resumen al final del procesamiento de términos del archivo invertido usando los parámetros *k1* y/o *k2*.

Ejemplo:

```
wwwisis db=cds k1=a k2=azzzzz pft=v1/ notermtail
```

out=<file> ; output file name outapp=<file> ; append output file name

El <file> contiene un nombre de archivo donde WWWISIS escribirá la salida. Cuando se especifica el parámetro *out* o *outapp*, WWWISIS no escribe la salida en la salida estándar del sistema y por tanto no se envía nada de vuelta al visualizador. Si <file> no existe, *out* y *outapp* son equivalentes, pero si <file> existe, *out* lo inicializará mientras que *outapp* agregará la salida.

Ejemplo:

```
wwwisis db=cds from=1 pft=@cd.pft out=cds.lst
```

pass=<user_password> ; user password

El <user_password> representa una clave de acceso de usuario que es agregada en el registro de entrada en el campo de información 1042. La especificación de formato en el parámetro *pft* puede chequear el <user_password> contra una base de datos.

Ejemplo:

```
wwwisis db=userdb from=1 count=1 user=ELISA pass=EVOLA pft=@chkuser
```

donde *chkuser* puede contener la siguiente especificación de formato:

```
if ref (l(mpu,v1041), mpu,v2) <> s(mpu,v1042) then 'User password  
invalid!'fi
```

(note que el *userdb* contiene la clave de acceso de usuario en el campo de tag 2 y la identificación de usuario invertida)

pend=<dbname> ; pending update data base name

El <dbname> representa el nombre de un archivo maestro de ISIS alternativo donde WWWISIS escribirá el registro nuevo o actualizado cuando la base de datos de entrada especificada en el parámetro *db* esté bloqueada.

El código de retorno de una operación de actualización se almacena en el campo de información de proceso con tag 1102. Cuando el código de retorno está entre 1 y 5, WWWISIS escribe el registro nuevo o actualizado en la base de datos pendiente. Cuando falla también la escritura en la base de datos pendiente, el código de salida del WWWISIS es 12 (almacenado en el registro log, campo de datos con tag 1091).

Los registros son agregados a la base de datos pendiente. Un proceso especial debe reunir los registros de las bases de datos pendientes y agregarlos a la base de datos de entrada. El campo de datos con tag 1103 contiene el MFN del registro en la base de datos de entrada (si el registro actualizado pudiera contener el campo de tag 1103, use el parámetro *tum* para asignar un número de tag diferente).

```
wwwisis cgi=@genparm.cgi
```

Donde *genparm.cgi* puede contener:

```
`db=', (if v100^n = `base' then v100^v fi),/  
'new',/  
'proc=', "a{"v1"{"", "a{"v2"{"", "a{"v3"{"", /,  
'user=', (if v100^n = `user' then v100^v fi), /,  
'pend=', (if v100^n = `user' then v100^v fi), /,  
'pft=@newrec.pft', /  
'prolog=@base.prl', /  
'epilog=@base.epi', /
```

En este ejemplo, el nombre de la base de datos pendiente es idéntico al nombre del usuario que es transferido a través del campo de entrada *user* .

pfxtag=<text_id> ; tag prefix to transfer values

El <text_id> activa el esquema de asignación de tags de WWWISIS basado en el contenido del atributo *name* de HTML del campo de entrada. La presencia del parámetro *pfxtag* sólo tiene sentido cuando está presente el parámetro *cgi*. Como *pfxtag* dirige la forma en que el parámetro *cgi* aplica el esquema de asignación de tags, no puede ser generado por el formato *cgi*, o sea, *pfxtag* debe aparecer codificado en la línea de comando antes del parámetro *cgi*. Ejemplo:

```
wwwisis pfxtag=tag cgi=@genparm.cgi
```

Cuando está presente el parámetro *pfxtag*, WWWISIS chequea el contenido de cada atributo *name* de campo de entrada para asignar un tag específico. Si *name* contiene el mismo prefijo especificado en el parámetro *pfxtag*, seguido por un número, este será el

número de tag del registro de datos en el que *value* será almacenado en el registro virtual de ambiente CGI; de lo contrario, si *name* contiene el identificador de subcampo *^t* seguido por un número, este será el número de tag en que será almacenado *value* en el registro virtual de ambiente CGI.

El *^r* seguido por un código ASCII puede ser agregada en el atributo *name* para indicar el carácter ASCII que va a separar las ocurrencias de un campo repetitivo.

Puede agregarse *^w* en el atributo *name* para indicar que el campo debe ser almacenado directamente en los registros actualizados y no en el registro virtual de ambiente CGI. El indicador *^w* puede usarse junto con *^t* en cuyo caso puede o no seguirse con un número de tag.

pft=<fmt> ; display format specification

El <fmt> representa una especificación in-line de formato o un nombre de fichero, precedido por el carácter @, y conteniendo una especificación de formato.

La especificación de formato de salida es obligatoria en cada llamado a WWWISIS. Puede ser especificada a través del parámetro *pft* (o los parámetros *pfttag* o *pftgen*). Acepta todos los comandos de especificación de formato de CDS/ISIS 3.07, excepto las salidas de formato (format exits). Además, pueden usarse varias extensiones de formato de la Biblioteca CISIS (Ver Anexo 1). Cuando <fmt> es igual a *ALL*, WWWISIS muestra el contenido de todos los registros.

El resultado del formato es enviado a la salida estándar. En un ambiente CGI, el servidor WWW lee la salida, y la envía a su vez al visualizador cliente. Este mecanismo permite la generación de textos en formato HTML de forma dinámica. En general, los elementos y atributos HTML se definen usando literales condicionales e incondicionales. Pero pueden ser también incluidos en el contenido de los campos de datos en los casos de bases de datos de texto completo. En particular, pueden desarrollarse aplicaciones de búsqueda interactiva usando el formato para generar formas HTML; cuando la interacción involucra varios estados diferentes, es necesario usar campos de entrada de FORM ocultos para transferir los datos entre estados.

Ejemplo:

```
wwwisis db=cds from=1 pft=ALL
```

```
wwwisis db=cds to=8 pft=@cdsx.pft
```

donde *cdsx.pft* contiene:

```
v24*0.20,c26,v70[1](25,25),if nocc(v70)>1 then ' et all' fi/
```

```
wwwisis db=cds to=8 pft=@html epilog='</table></body><html>'
```

donde *html* contiene:

```

if val(v1001) = 1 then
  '<html>' /
  '<body>' /
  '<br><table border=1>' / ,
fi,
'<tr><td valign=top>',v24,
'</td><td valign=top>',|<BR>|+v70
'</td></tr>' /

```

pftgen=<fmt>

El <fmt> representa una especificación in-line de formato o un nombre de fichero, precedido por el carácter @, y conteniendo una especificación de formato. El resultado del formato es la especificación real de formato para formatear los registros recuperados. Los parámetros *pfttag*, *pft* y *pftgen* son excluyentes.

Ejemplo:

```
wwwwisis db=cds pft=@onfly.pft from=1
```

donde *onfly.pft* puede contener:

```

'f(mfn,1,0),',
if v10 : '1' then
  '@format1.pft,'
else
  '@formatx.pft,'
fi,

```

pfttag=<tag_field> ; tag with display format specification

El <tag_field> representa un campo de datos del registro actual que contiene una especificación de formato. Los parámetros *pfttag*, *pft* y *pftgen* son excluyentes. El parámetro *pfttag* permite el uso de formatos específicos para registros individuales.

Ejemplo:

```
wwwwisis db=cds from=1 proc='a750{v24/{' pfttag=750
```

posting **posting=<n>**

La presencia del parámetro *posting* al recuperar registros usando los parámetros *k1* y/o *k2*, permite el acceso al posting de cada término recuperado del archivo invertido. Si el argumento <n> está presente, indica el número máximo de postings a procesar por término recuperado.

Cada término recuperado del archivo invertido se almacena en el campo de tag 1 de un registro de archivo maestro ISIS y el número correspondiente de postings en el campo de tag 2. WWWISIS recorre todos los postings de cada término y almacena los elementos del posting en el campo de tag 3 como subcampos. Este registro se mantiene sólo en memoria y siempre tiene Mfn igual a 1.

Los elementos del posting se describen en la tabla siguiente:

Subcampo	Descripción
\wedge j	Número de orden del posting
\wedge m	Mfn
\wedge t	ID de línea FST
\wedge o	Ocurrencia
\wedge c	Posición

Ejemplo:

```
wwwi32 db=cds k1=w k2=wzzz pft=v1,x1,v2,c40,v3(40,40)/ posting=3
```

proc=<fmt_proc> ; CISIS field update specification

El <fmt> representa una especificación in-line de formato o un nombre de fichero, precedido por el carácter @, y conteniendo una especificación de formato. La salida del formato es interpretada como una especificación de actualización de campo CISIS.

Una especificación de actualización de campo CISIS es una cadena de comandos de campo *d* (delete) y *a* (add) a ser aplicada en el registro actual. WWWISIS realiza los cambios en memoria. Para salvarlos en la base de datos, use los parámetros de actualización. Están disponibles los siguientes comandos:

D*	borra todos los campos presentes
Dtt	borra todas las ocurrencias del tag <i>tt</i>
Dtt/occ	borra la ocurrencia <i>occ</i> del tag <i>tt</i>
Att#str#	agrega la cadena <i>str</i> como nueva ocurrencia de <i>tt</i>
Htt n str	agrega la cadena <i>str</i> , de <i>n</i> bytes, como nueva ocurrencia de <i>tt</i>

Ejemplos:

```
wwwwisis db=cds "proc='d70',|a10#|v70|#|" pft=(v70/)
```

borrará el campo 70 y agregará el contenido al campo 10, o sea reenumerará el campo 70 como campo 10. El delimitador # puede ser cualquier carácter no numérico.

```
wwwwisis db=cds "proc=if v24*0.4 = 'Tech' then 'D*' fi" pft=v24/
```

borrará todos los campos si los primeros 4 caracteres del campo 24 son iguales a 'Tech'.

prolog=<fmt> ; cgi and environment format

El <fmt> representa una especificación in-line de formato o un nombre de fichero, precedido por el carácter @, y conteniendo una especificación de formato. La salida del formato es enviada directamente a la salida estándar. La especificación de formato actúa sobre el registro virtual de ambiente CGI que WWWISIS crea cuando comienza. También es posible formatear los registros de diferentes bases de datos físicas usando los comandos de formato *ref* (*reference*) y *l* (*look-up*) (ver Anexo 1).

El parámetro *prolog* se usa comunmente para generar el encabezamiento del texto en la salida estándar.

Ejemplo:

```
wwwisis prolog=@head.pft db=cds from=1 pft=mfn/
```

donde *head.pft* contiene la siguiente especificación de formato:

```
'<html> <header> <title> Example of WWWISIS output</title>  
</header> <body>'
```

quitpft=<fmt> ; quit pft loop

El <fmt> representa una especificación in-line de formato o un nombre de fichero, precedido por el carácter @, y conteniendo una especificación de formato. Si se produce alguna salida, WWWISIS interrumpe la ejecución.

Ejemplo:

```
wwwisis db=cds from=1 pft=@cds.pft "quitpft=if mfn>10 then 'x' fi"
```

read=<lock_id> ; read record lock id

El <lock_id> representa una identificación de bloqueo que se almacena en el campo de información de tag 1101 cuando se lee un registro y se bloquea exitosamente para ser actualizado en la corrida actual o en una corrida posterior de WWWISIS. El <lock_id> se usa en el parámetro *write* cuando se actualiza el registro en la base de datos. WWWISIS supone que toda actualización se realiza en un ambiente multiusuario.

se recomienda chequear el código de retorno de la operación de lectura. Los códigos de retorno de actualización de registros se almacenan en el campo 1102 y pueden contener los siguientes valores:

- 0 - Exito
- 1 - Bloqueo para entrada de datos negado
- 2 - Bloqueo de registro negado
- 3 - Actualización de registro negada
- 4 - Registro no bloqueado

- 5 - Identificación de bloqueo no coincide
- 6 - Actualización de invertido negada
- 7 - No se llegó al tiempo de expiración del bloqueo

Ejemplo:

```
wwwisis db=cds from=1 read=X pft=@chkrc
```

donde *chkrc* puede contener:

```
if val(v1102) > 0 then 'Record was not read! return code=', v1102
fi
```

reread ; allow same lock_id read again

La presencia del parámetro *reread* permite que el WWWISIS relea un registro que ya ha sido bloqueado, siempre que el argumento id de bloqueo coincida con el de actualización de WWWISIS almacenado previamente en el registro. El parámetro *reread* es útil cuando se solicita nuevamente el comienzo de la operación de actualización de un registro.

Ejemplo:

```
wwwisis db=cds read=sz reread from=10 count=1 pft=@cdsu.pft
```

reverse ; display on reverse sequence

La presencia del parámetro *reverse* dirige WWWISIS para procesar registros comenzando a partir del más alto Mfn o término del archivo invertido recuperado.

Ejemplo:

```
wwwisis db=cds from=1 pft=mfn(3),'-' reverse
```

```
wwwisis db=cds from=150 pft=mfn(3),'-' reverse
```

```
wwwisis db=cds from=1 count=10 pft=mfn(3),'-' reverse
```

```
wwwisis db=cds from=1 to=10 pft=mfn(3),'-' reverse
```

```
wwwisis db=cds k1=w k2=n step=100 reverse pft=v1/
```

step=<n> ; display every n

La <n> representa una tasa de muestra de los registros recuperados del archivo maestro o términos del archivo invertido a procesar. El valor por defecto es 1.

Ejemplo:

```
wwwisis db=cds from=1 step=10 pft=mfn/
```

```
wwwisis db=cds k1= step=10 pft=v1/
```


stop=<fmt> ; semaphore format

El <fmt> representa una especificación in-line de formato o un nombre de fichero, precedido por el carácter @, y conteniendo una especificación de formato. La salida del formato se envía directamente a la salida estándar. La especificación de formato actúa sobre el registro virtual de ambiente CGI que WWWISIS crea cuando comienza. También es posible formatear registros de diferentes bases de datos físicas usando los comandos de formato *ref* (*reference*) y *l* (*look-up*) (ver Anexo 1).

Si el formato produce cualquier salida, WWWISIS interrumpe la ejecución. El parámetro *stop* se usa con función de semáforo. De esta forma, puede chequear diferentes condiciones antes de la ejecución real, como por ejemplo, nombre e identificación de usuario, disponibilidad del sistema, consistencia de la entrada de datos, etc. Ejemplo:

```
wwwisis stop=@keyfmt.stp cgi=@genparm.cgi
```

El *keyfmt.stp* puede contener:

```
,if s(ref(['user']l(['user'],mpu,v1041),v2)) <> s(V1042) then
, 'User login or password incorrect!. Execution interrupted!'
, else,
,if s(ref(['semaph']l,v1)) = 'red' then
, 'System not available! Try later!'
,fi,
,fi,
```

t..=<tag> ; reassign information field tag

El <tag> representa un nuevo tag a asignar al campo de datos de información de proceso de WWWISIS identificado por el parámetro *t.* La siguiente tabla identifica los campos de información, los tags por defecto y los parámetros correspondientes para reasignar los tags:

Campo de Información	Parámetro	Tag por defecto
Current hit position tag	tpo	1001
Total records tag	ttt	1002
Reverse current hit position tag	trp	1003
First hit tag	tt1	1031
Last hit tag	tt2	1032
Previous hit MFN tag	tpm	1011
Search expression tag	tbo	1021
User identification tag	tui	1041
Password tag	tpa	1042
Data base name tag	tdb	1051
First term tag	tt1	1031
Last term tag	tt2	1032
User identification tag	tui	1041
Password tag	tpa	1042

Data base name tag	tdb	1051
Current key tag	tck	1
Current key postings tag	tpt	2
Current posting data	tps	3
Current MFN tag	tpo	1001
Total records tag	ttt	1002
Reverse current MFN tag	trp	1003
First MFN tag	tt1	1031
Last MFN tag	tt2	1032
Previous MFN tag	tpm	1011
User identification tag	tui	1041
Password tag	tpa	1042
Data base name tag	tdb	1051
CGI transfer tag	tcg	100
Lock identification tag	tul	1101
Read/write lock status	tus	1102
Pending MFN tag	tum	1103

Ejemplo:

```
wwwisis tcg=2001 cgi=@genparm
```

to=<mfn> ; last MFN

El <mfn> representa el mayor MFN a procesar.

Ejemplos:

```
wwwisis db=cds from=10 to=13 pft=mfn/
```

```
wwwisis db=cds from=10 to=13 pft=mfn/ reverse
```

```
wwwisis db=cds bool=w$ from=5 to=7 pft=mfn/
```

uctab=<file> ; upper case table

El <file> representa un nombre de archivo que contiene una tabla de conversión de caracteres a mayúscula de ISIS, o sea, una tabla de 256 códigos de caracteres que tiene para cada código de carácter ASCII el carácter mayúscula correspondientes. La tabla de mayúsculas se activa cuando uno de los comandos de formato *upper case mode* (*mpu*, *mhu*, *mdu*) está presente, y para convertir términos del archivo invertido a mayúscula, durante la actualización del archivo invertido y también durante la búsqueda.

Ejemplo:

```
wwwisis db=cds uctab=/isis/menu/isisuc.tab new fst=cds.fst
```

La tabla alfabética por defecto del WWWISIS es igual a la distribuida con MicroISIS.

Al definir una nueva tabla de mayúsculas, use siempre tres dígitos para cada entrada de código de carácter ascii. Cada línea tiene 32 entradas. La siguiente tabla se aplica al conjunto de caracteres ANSI:

000	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015
016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031
032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047
048	049	050	051	052	053	054	055	056	057	058	059	060	061	062	063
064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079
080	081	082	083	084	085	086	087	088	089	090	091	092	093	094	095
096	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079
080	081	082	083	084	085	086	087	088	089	090	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	138	155	140	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
065	065	065	065	065	065	065	067	069	069	069	069	073	073	073	073
068	078	079	079	079	079	079	215	216	085	085	085	085	089	222	223
065	065	065	065	065	065	065	067	069	069	069	069	073	073	073	073
208	078	079	079	079	079	079	247	216	085	085	085	085	221	222	221

unlock=<seconds> ; elapse seconds for unlock

El <seconds> representa el número de segundos luego de los que un registro bloqueado con una identificación de bloqueo id será desbloqueado. La id de bloqueo del registro es almacenada por WWWISIS en el campo de información de tag 1101 cuando un registro es bloqueado exitosamente usando el parámetro *read*.

Se recomienda chequear el código de retorno de la operación unlock. Los códigos de retorno de actualización de registros se almacenan en el campo 1102 y pueden contener los siguientes valores:

- 0 - Exito
- 1 - Bloqueo para entrada de datos negado
- 2 - Bloqueo de registro negado
- 3 - Actualización de registro negada
- 4 - Registro no bloqueado
- 5 - Identificación de bloqueo no coincide
- 6 - Actualización de invertido negada
- 7 - No se llegó al tiempo de expiración del bloqueo

Ejemplo:

```
wwwisis db=cds from=1 unlock=3600 pft=@chkrc
```

donde chkrc contiene:

```
if val(v1102) = 0 then mfn,' unlocked! '/ fi
```

user=<user_code> ; user identification

El <user_code> representa una identificación de usuario que se agrega al registro de entrada en el campo de información. La especificación de format en el parámetro *pft* puede chequear el <user_code> contra una base de datos.

Ejemplo:

```
wwwisis db=cds from=1 count=1 pft=@chkuser user=ELISA
```

donde *chkuser* puede contener la siguiente especificación de formato:

```
if val(v1001) = 1 then
  if ref(['userdb']l(['userdb']v1041), 'ok' ) <> 'ok' then
    'User login invalid! ',
  fi,
fi
```

write=<lock_id> ; write record lock id

El <lock_id> representa una identificación que será chequeada contra la identificación de bloqueo almacenada en el campo de información de tag 1101 del registro actual. Si coinciden, WWWISIS tratará de bloquear el registro para entrada y si lo consigue el registro será actualizado. La identificación de bloqueo es agregada por el parámetro *read* en la misma o en una corrida previa de WWWISIS.

Se recomienda chequear el código de retorno de la operación de escritura. Los códigos de retorno de la actualización de registros se almacenan en el campo 1102 y pueden contener los siguientes valores:

- 0 - Exito
- 1 - Bloqueo para entrada de datos negado
- 2 - Bloqueo de registro negado
- 3 - Actualización de registro negada
- 4 - Registro no bloqueado
- 5 - Identificación de bloqueo no coincide
- 6 - Actualización de invertido negada
- 7 - No se llegó al tiempo de expiración del bloqueo

Ejemplo:

```
wwwisis db=cds from=1 write=X pft=@chkrc
```

donde *chkrc* puede contener:

```
if val(v1102) > 0 then 'Record was not updated! return code=',
v1102 fi
```