BIREME / OPS / OMS

Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud

Utilitarios CISIS - Manual de Referencia

Versión 5.2

São Paulo, SP - 2005-2006

Utilitarios CISIS - Manual de Referencia

Ficha Catalográfica

BIREME / OPS / OMS (Brasil)

Utilitarios CISIS - Manual de Referencia. / BIREME / OPS / OMS. São Paulo, SP : BIREME / OPS / OMS, 2005-2006.

212 p.

1. Manual del usuario. 2. Acceso a la información. 3. Sistemas de información. 4. Gerenciamento de información. 5. Salud Pública. 6. Servicios de salud . I. BIREME II. Título

Advertencia - La mención a las compañías y/o instituciones específicas o a ciertos productos no implica que estos sean apoyados o recomendados por BIREME / OPS / OMS, y no significa que haya preferencia en relación a otros de naturaleza similar, citados o no.

BIREME / OPS / OMS

Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud

Rua Botucatu 862 V. Clementino

Este documento fue producido con la Metodología para la Normalización de Documentos (NorDoc) desarrollada por BIREME.

Tabla de contenido

Abreviaturas utilizadas ..... **X**
Como usar este manual ..... **XII**
Prefacio **1**
Sobre BIREME 1
La Biblioteca Virtual en Salud (BVS) 2
Presentación **4**
CISIS - Interfaz 4
CISIS - Programas Utilitarios 5
Instalación de los utilitarios CISIS 8
Ejecución de los utilitarios 9
Convenciones de sintaxis 11
Utilitario MX **13**
Presentación 13
Introducción *13*
Descripción general *14*
Sintaxis 18
Parámetros. Descripción general *20*
Parámetros de inicialización (setup) *20*
Parámetros que indican la fuente de entrada de datos *20*
Parámetros para procesamiento de datos *20*
Parámetros para selección de registros 20
Parámetros que realizan procesos 21
Parámetros de salida de datos 21
Parámetros generales 21
Parámetros que indican cuál es la fuente de entrada de datos 22
Base de datos de entrada *22*
Archivo ISO-2709 de entrada *23*
Largo de línea fijo 24
Lectura de archivos MARC 25
Datos del Leader del registro MARC 25

Archivo de texto ASCII de entrada *25*
Base de datos ficticia *29*
Archivo de parámetros *30*
Archivo Invertido como entrada *32*
Parámetros que realizan procesos sobre la entrada 34
Parámetros que aplican formatos a la entrada *34*
Especificación del formato de visualización en la línea de comando 35
Especificación del formato de visualización mediante un archivo 35
Formatos condicionales 36
Ancho de línea (*line width*) 36
Extrae datos del contenido de una variable CGI 37
Obtiene un archivo temporario vacío 37
Parámetros que seleccionan el conjunto de registros a ser procesados *38*
Especificación de la expresión de búsqueda en la línea de comando 38
Cargar la expresión de búsqueda desde un archivo 39
Utilización de los resultados intermedios de una búsqueda 40
Eliminación de la estadística de la búsqueda 42
Búsqueda en varios archivos invertidos 43
Búsquedas en texto libre 45
Parámetro text/show 45
Otras formas de seleccionar el conjunto de registros a procesar *46*
Selección por rango 46
Selección cada *n* registros 47
Seleccionar *n* registros 48
Parámetros que modifican registros *48*
Función 'A' (alta de campo) att#str# 52
Función R<mf>,<mfn> 55
Función <TAG[<stripmarklen>[<minlen>]]><data></TAG> 55
Función X[{create|copy|append|merge}=]<mf> 56
Función G<gizmo_mf>[,<taglist>] 57
Función Gsplit[/clean]=<tag>[={<char>|words|letters|numbers|trigram}] 57
Función Gsplit=<tag>=6words[/if=<if>] 58
Función Gload[/<tag>][/nonl][=<file>] 59
Función Gdump[/<tag>][/nonl][/xml][=<file>] 59
Cambio global de patrones *59*
Descripción 62
Tablas de conversión mediante códigos ASCII o hexadecimales 63
Estadística de la conversión por gizmo 65
Opción [decod=<mf>] 65
Unir bases de datos - JOIN 66
Descripción *67*
Lista de selección y renumeración de campos <tags> *72*
Contenido de los campos de control (32001, 32002, etc.) *73*
Join por número de registro *75*
Parámetro [jmax=<n>] *76*
Confrontar Bases de datos con archivos invertidos *76*
Contenido de los campos de control (32001, 32002, etc.) 79
Ventajas del *jchk* respecto del *join* 80
Tablas para conversión de caracteres *80*
Tablas para definición de caracteres alfanuméricos *80*
Tablas para conversión de caracteres alfabéticos a mayúsculas *81*

Tabla de Selección de Campos - generación de claves - fst *81*
Referencia a una tabla de selección de campos externa 82
Archivo de palabras no significativas (stopwords) 83
Técnicas de indización 1-8 / 1000 - 1008 85
Generación de archivos de ligas (links) 86
Archivos de ligas de longitud fija 87
Salida 88
Ejecución de programa externo *88*
Opción */show* 89
Parámetros que crean/modifican bases de datos *90*
Creación de un archivo maestro 90
Copiar registros a un archivo maestro 91
Agregar registros a una base de datos 92
Mezclar/Intercalar registros 92
Actualización de campos 93
Generar un Archivo ISO_2709 95
Generar un Archivo ASCII con separadores 96
Intercambiar datos del Leader del registro 97
Cargar elementos generados por una FST *98*
Funcion fullinv 98
Tabulación de frecuencia *99*
Parámetros de inicialización / variables de ambiente (setup) 100
Archivo de parámetros CISIS *100*
Tamaño máximo de un registro *101*
Tamaño máximo para el resultado de un formatoo de un formato *101*
Parámetros generales 102
Parámetros que controlan la salida por pantalla *102*
Parámetro + ... 102
Parámetro - 102
Parámetros para ambientes multiusuarios *104*
Opciones de proceso 104
Otros Parámetros *105*
Delimitadores 105
Indicadores (*prompts*) predeterminados 105
Parámetro *trace* 106
Parámetro *mfrl* 107
MX: código de retorno de ejecución 107
Utilitarios del archivo maestro 108
MXF0 - Programa 108
MXF0 - Presentación 109
MXF0 - Sintaxis 110
Parámetros obligatorios *110*
Nombre del archivo maestro de entrada 110
Nombre del archivo maestro de salida 110
Crear archivo maestro de salida 111
Cantidad aproximada de registros 111
Parámetros opcionales *111*
Eliminación de espacios en blanco 111
Información sobre la ejecución del proceso 112
MXF0 - Salida 112
MXTB - Programa 113

MXTB - Presentación 113
MXTB - Sintaxis 115
Parámetros obligatorios 116
Nombre del archivo maestro de entrada 116
Nombre del archivo maestro de salida 116
Crear archivo maestro de salida 117
Clave 117
Longitud máxima de la clave 117
Formato que especifica la clave 117
Parámetros opcionales 118
Procesar el resultado de una búsqueda 118
Tabulación del resultado de formato 118
Cantidad de categorias 119
MXTB - Salida 119
MXCP - Presentación 120
MXCP - Sintaxis 123
Nombre del archivo maestro de entrada 123
Nombre del archivo maestro de salida 123
Crear archivo maestro de salida 124
Parámetros opcionales [option] 124
Recuperar registros borrados 124
Cambio global de patrones 125
Convertir campos en repetibles 126
Supresión de espacios en blanco 126
Eliminación de campos por tag 126
Registro de eventos 127
MXCP - Salida 127
MSRT - Programa 128
MSRT - Presentación 128
MSRT - Sintaxis 129
Parámetros obligatorios: 129
Nombre del archivo maestro de entrada 129
Longitud máxima de la clave 129
Generación clave 129
Parámetros opcionales 130
Mantener MFNs originales 130
Borrar claves iguales 130
MSRT - Salida 130
RETAG - Programa 131
RETAG - Presentación 131
RETAG - Sintaxis 132
Parámetros obligatorios 132
Archivo maestro de entrada 132
Tabla de renumeración 132
Parámetros opcionales 133
RETAG - Salida 133
CTLMFN - Programa 133
CTLMFN - Presentación 134
CTLMFN - Sintaxis 135
Nombre del archivo maestro de entrada 135
Prompt de confirmación 135

CTLMFN - Salida 135
MKXRF - Programa 135
MKXRF - Presentación 136
MKXRF - Sintaxis 137
Archivo maestro de entrada 137
MKXRF - Salida 137
Restaurando una base de datos dañada 138
Cálculo aproximado del MaxMFN 138
ID2I - Programa 138
ID2I - Presentación 139
Estructura del archivo ASCII: 139
ID2I - Sintaxis 140
Parámetros obligatorios 140
Archivo ASCII de entrada 140
Nombre del archivo maestro de salida 140
Crear archivo maestro de salida 140
Parámetros opcionales 141
I2ID - Programa 141
I2ID - Presentación 141
I2ID - Sintaxis 142
Parámetros obligatorios: 142
Archivo maestro de entrada 142
Parámetros opcionales 142
CRUNCHMF - Sintaxis 143
Utilitarios del archivo invertido 144
IFKEYS - Programa 144
IFKEYS - Presentación 144
IFKEYS - Sintaxis 145
Archivo invertido de entrada 145
Parámetros opcionales 146
Primer término a ser listado 146
Ultimo término a ser listado 146
Mostrar información sobre etiquetas (tags) 146
IFKEYS - Salida 146
IFLOAD - Programa 147
IFLOAD - Presentación 147
IFLOAD - Sintaxis 150
Parámetros obligatorios 150
Archivo invertido de entrada 150
Archivo de ligas de claves cortas 150
Archivo de ligas de claves largas 150
Parámetros obligatorios 151
Reinicializar marca de actualización pendiente 151
Actualización pendiente 151
No balancear diccionario 151
No Cargar postings 151
Información sobre la ejecución del proceso 152
Archivos de ligas de longitud fija 152
Cargar archivos de longitud fija 152
Carga archivo de ligas con formato reducido 152
IFLOAD - Salida 152

IFUPD - Programa 153
IFUPD - Presentación 153
IFUPD - Sintaxis 154
Parámetros obligatorios 155
Archivo invertido a ser actualizado 155
Tabla de selección de campos 155
FST por defecto 155
Cargar FST desde un archivo externo 155
Especificación de FST en línea 155
Archivo de palabras no significativas 156
Archivo STW por defecto 156
Cargar lista STW desde un archivo externo 156
Mantener marca de actualización pendiente 156
No cargar postings 156
Archivo maestro alternativo 157
IFUPD - Salida 157
MYS - Sintaxis 157
MYS – Salida 157
IFMERGE - Sintaxis 158
IFMERGE – Salida 158
MKIY0 - Sintaxis 158
MKIY0 - Salida 158
CRUNCHIF - Sintaxis 158
CRUNCHIF - Salida 159
Referencias bibliográficas 160
Glosario 161
Apéndice I - Parámetros de uso general 164
Relativos a la salida estándar: 164
Deshabilitar prompt entre registros 164
Informar cada n registros 165
Deshabilitar volcado de información en pantalla 165
Redireccionar salida estándar 166
Relativos a la selección de registros: 167
Iniciar en registro n 167
Finalizar en registro n 167
Procesar un registro cada n 167
Seleccionar n registros 167
Relativos a los registros de salida: 168
Sumar n a los números de registro 168
Cambio global de patrones 169
Apéndice II - Archivo CIPAR 172
Parámetros que se pueden incluir en el CIPAR 175
Parámetros sólo para MX 175
Parámetros para MX y aplicaciones programadas para ambiente multiusuario 178
Parámetro maxmfrl 179
Parámetro mstxl en el CIPAR 179
Cómo superar el límite de 512 MB para el archivo maestro: 179
Parámetro dbxtrace=y 180
Parámetro mstload=<n> 181
Parámetro invload=<n> 181
Parámetro mclose={y|n} 181

Parámetro iflush={y|n} 181
Parámetro mflush={y|n} 182
Parámetro what={y|n} 182
Apéndice III - Estructura de los registros de una base ISIS 189
El Registro de CONTROL 190
El Registro de XREF 191
El Registro del Archivo MST 191
Estructura del LEADER 192
Estructura del DIRECTORIO 192
Apéndice IV - Lista de archivos TABs disponibles 193
ASCII CODE PAGE 437 (CP437) 193
ASCII CODE PAGE 850 (CP850) 193
ANSI (Windows) 194
GIZMOs disponibles para conversión del contenido de bases de datos 194
Conversión del conjunto de caracteres 194
ASCII CODE PAGE 437 (CP437) 194
ASCII CODE PAGE 850 (CP850) 195
ANSI (Windows) 195
Conversión auxiliar de caracteres de marcación 195
Conversión auxiliar de y para entidades HTML 195
Cómo reconocer el conjunto de caracteres en que está una base CDS/ISIS 195
OBSERVACIONES 196
Apéndice V - MX.PFT: Lista de parametros que extrae del environment de CGI 197

Abreviaturas utilizadas

- **ANSI**. American National Standards Institute [Instituto Nacional Americano de Normas].
- **ASCII**. American Standard Code for Information Interchange [Código Americano Normalizado para el Intercambio de Información].
- **BIREME**. Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud.
- **BVS**. Biblioteca Virtual en Salud.
- **CDS**. Computerized Documentation System [Sistema de Documentación Computarizada].
- **CP**. Code Page [Código de página].
- **FST**. Field Selection Table [Tabla de Selección de Campo].
- **FTP**. File Transfer Protocol [Protocolo de transferencia de archivos].

- **IFP**. Inverted File Pointer [Puntero de archivo invertido].
- **ISIS**. Integrated Set of Information Systems [Conjunto integrado de sistemas de información].
- **ISO**. International Organization for Standardization [Organización Internacional para la Normalización].
- **LILACS**. Literatura Latinoamericana y del Caribe en Ciencias de la Salud.
- **OMS**. Organización Mundial de la Salud.
- **OPS**. Organización Panamericana de la Salud.
- **UNESCO**. United Nations Educational, Scientific and Cultural Organization [Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura].

Como usar este manual

Este manual fue concebido como una referencia para el uso del los utilitarios CISIS y su contenido principal está dividido en cuatro capítulos, siendo:

1. Presentación: la descripción del CISIS, de la interfaz, instalación, ejecución y convenciones utilizadas por los utilitarios;
2. Utilitario MX: conteniendo la descripción completa de todos los parámetros y funciones disponibles;
3. Utilitarios del archivo maestro: describe la sintaxis y uso de los programas utilitarios para archivos maestros.
4. Utilitarios del archivo invertido: describe la sintaxis y uso de los programas utilitarios para archivos invertidos.

Además hay cuatro apéndices conteniendo información adicional transversal a los capítulos y también información acerca de la estructura interna del ISIS.

Un *Glosario* y una lista de *Abreviaturas utilizadas* complementan el documento.

Prefacio

Sobre BIREME

Año tras año, BIREME cumple su misión como centro especializado en información científica y técnica en salud para la región de América Latina y el Caribe. Establecida en Brasil en 1967, con el nombre de Biblioteca Regional de Medicina (que originó la sigla BIREME), atendió desde el inicio a la creciente demanda de literatura científica actualizada por parte de los sistemas nacionales de salud y las comunidades de investigadores, profesionales y estudiantes. Posteriormente, en 1982, pasó a llamarse Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud, para mejor expresar sus funciones, orientadas al fortalecimiento y ampliación del flujo de información científica y técnica en salud en toda la región, pero conservó su sigla.

El trabajo en red, en base a la descentralización, orientado a desarrollar capacidades locales, compartir recursos de información, desarrollar productos y servicios cooperativos, elaborar metodologías comunes, siempre fue el fundamento del trabajo de cooperación técnica de BIREME. De esa forma el centro se consolida como un modelo internacional que promueve la capacitación de los profesionales de información a nivel gerencial y técnico, para que adopten los paradigmas de información y comunicación que mejor atiendan a las necesidades locales.

Los principales fundamentos que dan origen y soporte a la existencia de BIREME son los siguientes:

- el acceso a la información científico-técnica en salud es esencial al desarrollo de la salud;
- la necesidad de desarrollar la capacidad de los países de América Latina y el Caribe de operar las fuentes de información científico-técnica en salud de forma cooperativa y eficiente;
- la necesidad de promover el uso y de responder a las demandas de información científico-técnica en salud de los gobiernos, los sistemas de salud, las instituciones de enseñanza e investigación.

BIREME, como centro especializado de la Organización Panamericana de la Salud (OPAS)/Organización Mundial de la Salud (OMS), coordina y realiza actividades de cooperación técnica en gestión de información y conocimiento científico, con el propósito de fortalecer y ampliar el flujo de información científica en salud en Brasil y en los demás países de América Latina y el Caribe, como condición esencial para el desarrollo de la salud, incluyendo planificación, gestión, promoción, investigación, educación y atención.

El convenio que fundamenta BIREME es renovado a cada cinco años por los miembros del Comité Asesor Nacional de la institución (OPAS, Ministerio de la Salud de Brasil, Ministerio de Educación y Cultura de Brasil, Secretaría de Salud del Estado de São Paulo y Universidad Federal de São Paulo – Unifesp). Esta última ofrece la infraestructura física necesaria al establecimiento de la institución.

En 2004 la institución asumió la responsabilidad de convertirse en una entidad que se basa en el conocimiento.

La Biblioteca Virtual en Salud (BVS)

Con el surgimiento y consolidación de la Internet como medio predominante de información y comunicación, el modelo de cooperación técnica de BIREME evolucionó desde 1998 hacia la construcción y desarrollo de la Biblioteca Virtual en Salud (BVS) como espacio común de convergencia del trabajo cooperativo de productores, intermediarios y usuarios de información. La BVS promueve el desarrollo de una red de fuentes de información científica y técnica con acceso

universal en la Internet. Por primera vez se abre la posibilidad real de acceso equitativo a la información en salud.

BIREME tiene a la Biblioteca Virtual como modelo para la gestión de información y conocimiento, lo que implica la cooperación y convergencia de instituciones, sistemas, redes e iniciativas de productores, intermediarios y usuarios en la operación de redes de fuentes de información locales, nacionales, regionales e internacionales, privilegiando así el acceso abierto y universal.

Actualmente, todos los países de América Latina y el Caribe (Región) participan directa o indirectamente en los productos y servicios cooperativos promovidos por la BVS, lo que involucra a más de mil instituciones en más de 30 países.

La BVS es simulada en un espacio virtual de la Internet formada por la colección o red de fuentes de información en salud de la Región. Usuarios de distintos niveles y localización pueden interactuar y navegar en el espacio de una o varias fuentes de información, independientemente de su localización física. Las fuentes de información son generadas, actualizadas, almacenadas y operadas en la Internet por productores, integradores e intermediarios, de modo descentralizado, obedeciendo a metodologías comunes para su integración a la BVS.

La BVS organiza la información en una estructura que integra e interconecta bases de datos referenciales, directorios de especialistas, eventos e instituciones, catálogo de recursos de información disponibles en la Internet, colecciones de textos completos con destaque para la colección SciELO (Scientific Electronic Online) de revistas científicas, servicios de diseminación selectiva de información, fuentes de información de apoyo a la educación y la toma de decisión, noticias, listas de discusión y apoyo a comunidades virtuales. Por lo tanto, el espacio de la BVS constituye una red dinámica de fuentes de información descentralizada a partir de la cual se puede recuperar y extraer información y conocimiento para subsidiar los procesos de decisión en el área de la salud.

La Biblioteca Virtual en Salud es visualizada como la base distribuida del conocimiento científico y técnico en salud registrado, organizado y almacenado en formato electrónico en los países de la Región, accesible de forma universal en la Internet de modo compatible con las bases internacionales.

Presentación

CISIS - Interfaz

MicroISIS (*CDS/ISIS for Mini-microcomputers*) es un software desarrollado por UNESCO para bases de datos constituidas mayoritariamente por texto. MicroISIS maneja campos (elementos de datos) de longitud variable. Un campo puede estar ausente en uno o más registros, puede contener sólo un elemento de dato, o dos o más subcampos de longitud variable. Asimismo, un campo puede ser repetible, esto es, un registro dado puede contener más de una ocurrencia del campo.

La Interfaz CISIS es una biblioteca de funciones, escrita en lenguaje de programación C, diseñada para permitir el desarrollo de aplicaciones para bases de datos MicroISIS (sin llamar al software MicroISIS). Las aplicaciones CISIS son plenamente compatibles con MicroISIS, incluyendo aplicaciones multiusuario.

Existen diversas implementaciones sobre la estructura CDS/ISIS original, por lo que hoy día es más apropiado llamar a todas estas variantes como “familia Isis”. Es importante destacar que los datos creados bajo cualquiera de las variantes de esta “familia” son compatibles y pueden intercambiarse entre ellas.

Las aplicaciones desarrolladas con la Interfaz CISIS pueden manipular varias bases de datos al mismo tiempo; el archivo maestro y el archivo invertido son

procesados en forma independiente. No es necesario tener la definición de la base de datos para correr las aplicaciones CISIS.

La Interfaz CISIS y los Programas Utilitarios CISIS fueron diseñados e implementados en el Centro de Información en Ciencias de la Salud para América Latina y el Caribe - BIREME, Organización Panamericana de la Salud - OPS, y actualmente están disponibles bajo plataformas:

- PC IBM[1] 32 bits
- UNIX basados en procesadores Intel (LINUX, SCO, etc.)
- UNIX basados en otros procesadores (HP-UX, Sun, IBM-AIX, CDC/S4320, etc.)
- VAX bajo VMS
- HP3000/950 bajo MPE/XL

CISIS - Programas Utilitarios

Los Utilitarios CISIS (*CISIS Interface Utility Programs*) son un conjunto de programas desarrollados en lenguaje de programación C que "llaman" las funciones ofrecidas por la Interfaz CISIS para realizar distintas funciones sobre bases de datos de la familia Isis, tales como recuperar y mostrar registros, el mantenimiento de bases de datos, etc. Asimismo pueden efectuar funciones especiales que permiten ordenar archivos maestros, generar tablas a partir de un archivo maestro, cambiar las etiquetas de los campos, etc.

Este conjunto de programas utilitarios se ofrece bajo cuatro versiones: 10/30 y 16/60, LIND, FFI. Las diferencias sustanciales están en el largo de las claves del archivo invertido y el tamaño máximo de registro medido en bytes que soportan, según se muestra en la tabla siguiente.

[1] Para MS-DOS de 16 bits deberán usarse las aplicaciones de CISIS hasta la versión 3.4

	10/30	16/60	LIND	FFI
Claves archivo invertido	30	60	60	60
Tamaño máximo del registro	32.767	32.767	32.767	1.048.576

Nota: La versión 10/30 es la única compatible con el CDS/ISIS de Unesco

Para más detalles sobre la estructura de los archivos maestro e invertido vea el *Apéndice III - Estructura de los registros de una base ISIS.*

Las características particulares de estos programas, pueden verificarse en la declaración de versión que se obtiene con el comando what

Por ejemplo

```
mx what
CISIS Interface v5.2a/PC32/M/32767/10/30/I - Utility MX
CISIS Interface v5.2a/.iy0/Z/4GB/GIZ/DEC/ISI/UTL/INVX/B7/FAT/CIP/CGI/MX
Copyright (c)BIREME/PAHO 2006. [http://www.bireme.br/products/cisis]
```

Acrónimo	Descripción
V5.2a	número de la versión
PC32	Compilador usado (en este caso Windows PC)
L	Versión Lind si está presente
M	Multi-user support
32767	Tamaño máximo del registro en bytes, valor por defecto
10/30	largo de las claves del archivo invertido
I	Permite actualizar el I/F
Utility MX	Nombre del programa
.iy0	Single physical file for I/F (construido por mkiy0)
Z	Compressed I/F (construido por myz – descontinuado)
4GB	Tamaño máximo del archivo maestro
GIZ	Gizmo
DEC	Decod
ISI	Iso-2709 import
UTL	Ciutl module
INVX	Multiple I/F searching
B7	Versión del mecanismo interno de búsqueda
FAT	Fatal()
CIP	Cipar()
GCI	Soporta la operación en ambiente CGI
MX	Cisis_mx()

Utilitario MX

MX

El Programa MX es un utilitario de propósito general para trabajar sobre bases de datos MicroISIS. Puede realizar la mayoría de las funciones de la Interfaz CISIS, incluyendo la importación/exportación de archivos ISO-2709, búsquedas, procedimientos de cambio global de patrones, unión de registros del archivo maestro por número de registro o por clave del archivo invertido, incorporar campos con datos generados mediante una Tabla de Selección de Campos (FST), y funciones de alta y baja de campos.

Utilitarios para archivo maestro

MXF0

Analiza todos los registros de un archivo maestro dado, produciendo información acerca de los campos presentes y de los caracteres usados por éstos.

MXTB

El programa MXTB permite contar el contenido de los campos, por ejemplo, cantidad de veces que aparece cada autor, cada descriptor, o la aparición simultánea de un autor y un título de publicación periódica, etc.
El resultado de la ejecución de MXTB es un archivo maestro que contiene un registro por cada frase diferente hallada (categoría). Estos registros tienen campos para almacenar la categoría y su frecuencia.

MXCP

Copia registros desde un archivo maestro de entrada a un archivo maestro de salida, posibilitando que el dato de entrada sea modificado por procedimientos de cambio global de patrones y/o procedimientos que suprimen espacios al comienzo o al final, caracteres en blanco, caracteres no imprimibles y caracteres de puntuación final.
También convierte en repetibles los campos que contengan un delimitador específico y puede descartar campos de entrada, según los valores de sus tags (*etiquetas*).
Otra característica del MXCP es la de recuperar (*undelete*) registros lógicamente borrados del archivo maestro.

MSRT

Ordena los registros de un archivo maestro en forma ascendente, de acuerdo a claves que se generan aplicando un formato a los registros.

RETAG

Este programa tiene dos aplicaciones:
Cambiar los tags (*etiquetas*) de los campos de un archivo maestro dado, según una tabla de renumeración.
Desbloquear (*unlock*) un archivo maestro.

CTLMFN

Despliega y actualiza el registro de control del archivo maestro.
Debe usarse cuando un archivo maestro es reinicializado por accidente.

MKXRF

Es un programa para recuperación del archivo maestro, que lee un archivo *.mst* y crea el archivo *.xrf* correspondiente.
Puede ser usado para restaurar todos los registros activos en un archivo maestro reinicializado en forma lógica.

Utilitarios para archivo maestro

`I2ID`	Lee un archivo maestro y genera un archivo ASCII, que puede ser editado y modificado. La idea es que trabaje junto con el utilitario *ID2I* que realiza la tarea inversa: lee un archivo ASCII y convierte los datos leídos en registros de archivo maestro.
`ID2I`	Lee un archivo ASCII generado por *I2ID* (o con la misma estructura que un archivo generado por éste) y convierte los datos leídos en registros de archivo maestro.
`CRUNCHMF`	Convierte el archivo maestro de un sistema operativo a otro, por ejemplo de Windows a Linux.

Utilitarios para archivo invertido

`IFKEYS`	Despliega los términos del archivo invertido y la cantidad de *postings* de cada uno de ellos. Opcionalmente los términos pueden ser desagregados por las etiquetas de los que fueron extraídos.
`IFLOAD`	Carga un archivo invertido a partir de los archivos de ligas, según las opciones de procesamiento. Acepta otros formatos, además del formato de archivo de ligas estándar de MicroISIS.
`MYS`	Hace un sort del archivo de ligas (links) para crear el archivo invertido.
`IFMERGE`	Combina varios archivos invertidos de diferentes archivos master en un solo archivo invertido, con un procedimiento para recuperar los registros desde los archivos master fuentes.
`MKIY0`	Combina los seis archivos que componen el archivo invertido en un solo archivo físico.
`CRUNCHIF`	Convierte el archivo invertido de un sistema operativo a otro, por ejemplo de Windows a Linux.

Instalación de los utilitarios CISIS

Toda la instalación de los Utilitarios CISIS consiste en crear un directorio, por lo general \CISIS\SYS\, y copiar en éste todos los utilitarios.

Por una cuestión de comodidad se puede agregar el directorio \CISIS\SYS al PATH del sistema operativo, para poder ejecutar los utilitarios desde la ubicación en que uno se encuentre, sin tener que referenciar al directorio \CISIS\SYS.

Ejemplos del manual

Los ejemplos del manual están basados en su mayoría en la base de datos CDS, y se supone que está ubicada en el directorio:

```
\CISIS\BASES\
```

Se trabajará sobre la base de datos y muchas veces se la modificará, por lo que se aconseja realizar una copia de resguardo de la misma.

Ejecución de los utilitarios

Los programas utilitarios CISIS se ejecutan como comandos, desde el *prompt* del sistema operativo, o desde archivos *bat* (archivos de procesamiento por lotes) de MS-DOS o *scripts* (shell scripts) de UNIX.

Cualquier programa utilitario CISIS puede ser ejecutado escribiendo su nombre y uno o más parámetros, suponiendo que el directorio \cisis\sys (directorio donde se encuentran los utilitarios CISIS) esté incluido en la lista PATH. Si no se suministran parámetros en la llamada, cada programa utilitario CISIS despliega una descripción breve de su uso. Por ejemplo, escribiendo sólo el nombre MXCP en el *prompt* de DOS, se despliega:

```
CISIS Interface v5.2a/PC32/M/32767/10/30/I - Utility MXCP
Copyright (c)BIREME/PAHO 2006. [http://www.bireme.br/products/cisis]

mxcp {in=<file>|<dbin>} [create=]<dbout> [<option> [...]]

options: {from|to|loop|count|tell|offset}=<n>
         gizmo=<dbgiz>[,tag_list>]
         undelete
         clean [mintag=1] [maxtag=9999]
         period=.[,<tag_list>]
```

```
            repeat=%[,<tag_list>]
            log=<filename>

Ex: mxcp in create=out clean period=.,3 repeat=;,7

     in = 3 «   Field 3 occ 1. »
          3 «Field 3 occ 2 . »
          7 « Field 7/1;Field 7/2 ;Field 7/3.»

    out = 3 «Field 3 occ 1»
          3 «Field 3 occ 2»
          7 «Field 7/1»
          7 «Field 7/2»
          7 «Field 7/3.»
```

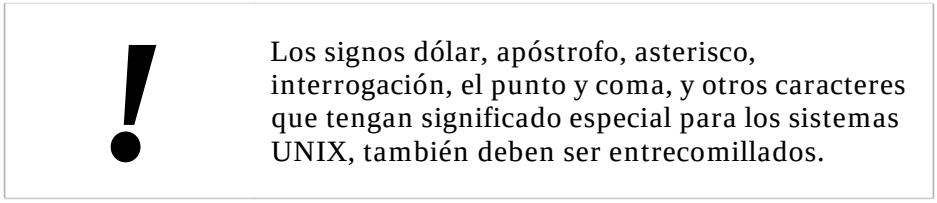
Los parámetros de la llamada se proveen como una lista separada por espacios en blanco y, por lo tanto, cada parámetro individual debe ser entrecomillado cuando contenga espacios en blanco o cualquier carácter especial del sistema operativo (tales como paréntesis angular, barra vertical, etc.).

El ejemplo siguiente ejecuta el programa MX con tres parámetros de llamada (nombre de la base de datos, expresión de búsqueda y especificación del formato de despliegue):

```
mx \cisis\bases\cds "plants*water" "pft=mfn/,'Ti: 'v24/,(|Au: |v70/)"
```

Para usar la doble comilla como parte de un parámetro de la llamada, debe estar precedida por la barra invertida:

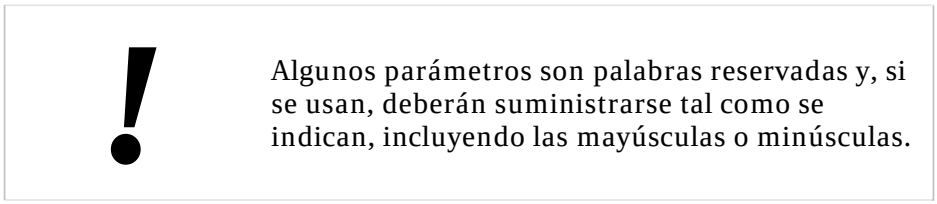
```
mx \cisis\bases\cds "plants*water" "pft=mfn/, \" Ti: \"v24/,(|Au: |v70/)"
```



Convenciones de sintaxis

Se usan las siguientes convenciones para describir la sintaxis de los Programas Utilitarios CISIS:

<parameter>	parámetro obligatorio
[<parameter>]	parámetro opcional
{<option 1>|<option 2>}	debe elegirse entre <opción 1> u <opción 2>
<option> [...]	<opción> puede repetirse



Algunos parámetros son palabras reservadas y, si se usan, deberán suministrarse tal como se indican, incluyendo las mayúsculas o minúsculas.

Por ejemplo, MXCP tiene la sintaxis general:

```
mxcp <dbin> [create=]<dbout> [<option> [...]]

options:

{from|to|loop|count|tell|offset}=<n>
gizmo=<dbgiz>[,<tag_list>]
undelete
clean [mintag=1] [maxtag=9999]
period=.[,<tag_list>]
repeat=%[,<tag_list>]
log=<filename>
```

Indicando que dos parámetros son obligatorios: (a) nombre de la base de datos de entrada y (b) nombre de la base de datos de salida.

Entonces, el comando:

```
mxcp \cisis\bases\cds newcds
```

Copia el archivo maestro *cds* ubicado en el directorio \cisis\bases al archivo maestro *newcds* situado al directorio actual. El archivo maestro *newcds* deberá existir, de lo contrario el ejemplo producirá un error.

Si *newcds* no existe puede crearse mediante el parámetro opcional *create* como se ve en el siguiente ejemplo:

```
mxcp \cisis\bases\cds create=newcds
```

Las opciones de proceso pueden indicarse usando los parámetros opcionales. Para indicar, por ejemplo, límites de registros a procesar se utiliza *from* y *to:*

```
mxcp \cisis\bases\cds create=newcds from=10 to=20
```

Utilitario MX

Presentación

Introducción

MX es un programa de uso general para bases de datos CDS/ISIS que realiza la mayoría de las funciones de la Interfaz CISIS. Al igual que los otros programas utilitarios CISIS, MX se ejecuta desde la línea de comandos del sistema operativo, indicando las operaciones a realizar mediante parámetros.

MX puede utilizarse, por ejemplo, para recuperar y mostrar un conjunto de registros de una base de datos, de acuerdo a una expresión de búsqueda y un formato de visualización, como en la siguiente línea:

```
mx \cisis\bases\cds "plants * water" "pft=mfn,x1,v24/"
```

Asimismo, MX permite realizar búsquedas en texto libre aunque no exista un archivo invertido.

MX también puede leer archivos ISO-2709 o archivos ASCII planos, utilizando delimitadores como separadores de campos. En estos casos los registros de entrada son convertidos a registros de archivo maestro a medida que son leídos.

Los siguientes procedimientos pueden ser aplicados a los registros de entrada:

1. Cambio global de patrones.
2. Unión de registros, por número de registro o por clave de archivo invertido.
3. Agregar campos con los datos generados por una Tabla de selección de campos.
4. Alta y baja de registros, especificadas a través de un lenguaje de formateo.

Los registros procesados por MX pueden enviarse a un archivo maestro, un archivo ISO-2709 o a la salida estándar (la cual puede ser redireccionada a un archivo o impresora). Las líneas producidas por un formato pueden ser enviadas al sistema operativo.

La ejecución del MX puede generar una llamada al sistema operativo para que éste realice determinado programa.

El resultado de aplicar una Tabla de Selección de Campos (FST) a un archivo maestro puede enviarse a un archivo de ligas (*link files*) o agregarse a un archivo invertido.

El archivo de salida puede ser el mismo que el de entrada.

MX trabaja también en entornos multiusuarios.

Descripción general

Para ejecutar el MX es necesario especificarle dónde están los datos sobre los que va a trabajar. Pueden provenir de un archivo maestro, un archivo ISO-2709 o un archivo de texto. Este es el único parámetro obligatorio del programa.

La línea

```
mx \cisis\bases\cds
```

genera un listado en pantalla de los registros de la base *cds*, que se encuentra en el directorio \cisis\bases. Los registros listados se visualizan sin formatear.

Otros parámetros pueden especificar opciones de proceso, por ejemplo:

```
mx \cisis\bases\cds from=10 to=20
```

Presentando en pantalla los registros de la base de datos *cds* desde el 10 hasta el 20. La base se encuentra en el directorio \cisis\bases y los registros se visualizan sin formatear.

La línea de comando

```
mx \cisis\bases\cds from=10 to=20 "pft=mfn,x1,v24(0,7)/"
```

despliega en pantalla los registros 10 a 20 de la base de datos cds, aplicándoles el formato especificado en el parámetro pft=mfn,x1,v24(0,7)/. La base de datos se encuentra en el directorio \cisis\bases.

Es importante tener en cuenta que el orden en que se disponen los parámetros opcionales **no afecta** a la ejecución del MX. La ejecución de estos parámetros se hará en el orden en que aparecen en la declaración de sintaxis.

Así la línea anterior podría ser:

```
mx \cisis\bases\cds   to=20   "pft=mfn,x1,v24(0,7)/"   from=10
```

o

```
mx \cisis\bases\cds   "pft=mfn,x1,v24(0,7)/"   from=10   to=20
```

Sin embargo, si bien la siguientes declaraciones son equivalentes:

```
mx \cisis\bases\cds   pft=@file1  proc=@miproc.prc
mx \cisis\bases\cds   proc=@miproc.prc  pft=@file1

mx \cisis\bases\cds   gizmo=gizfile1  proc=@miproc.prc
mx \cisis\bases\cds   proc=@miproc.prc  gizmo=gizfile1
```

el parámetro gizmo se aplicará antes que el proc, porque así se establce en la sintaxis.

Si la entrada principal (primer parámetro) es una base de datos y su correspondiente archivo invertido está disponible, el conjunto de registros a ser procesados pueden obtenerse mediante una búsqueda.

El siguiente ejemplo recupera los registros de la base *cds*, que se encuentra en el directorio \cisis\bases, que contengan las palabras plants y water.

```
mx \cisis\bases\cds "plants * water"
```

MX puede leer los datos de entrada desde un archivo con formato ISO-2709 o un archivo de texto con delimitadores.

La siguiente línea despliega los primeros 5 registros de un archivo ISO-2709 llamado cds.iso, que se encuentra en el directorio \cisis\bases.

```
mx iso=\cisis\bases\cds.iso to=5
```

En el próximo ejemplo el MX utiliza un archivo ASCII llamado *name* como fuente de entrada, cuyo contenido es:

```
Autor 1|título 1|^aParis^bUnesco^c1965
|título 2|^aParis^bUnesco^c1965
Autor 3|título 3|^aParis^bUnesco^c1965
```

Y puede ser listado mediante la siguiente llamada al MX:

```
mx seq=name "pft=mfn,c11,v1,c21,v2,c31,v3/" now
```

Que generará la salida:

```
000001    Autor 1   Title 1   ^aParis^bUnesco^c1965
000002              Title 2   ^aParis^bUnesco^c1965
000003    Autor 3   Title 3   ^aParis^bUnesco^c1965
```

Los registros procesados pueden almacenarse en un archivo maestro. Las siguientes líneas crean el archivo maestro *sample*:

```
mx\cisis\bases\cds "plants * water" create=sample -all now
mx iso=\cisis\bases\cds.iso to=5 create=sample -all now
mx seq=name create=sample -all now
```

Estos registros, también, pueden exportarse a un archivo con formato ISO-2709, por ejemplo *sample.iso:*

```
mx \cisis\bases\cds "plants * water" iso=sample.iso -all now
mx iso=\cisis\bases\cds.iso to=5 iso=sample.iso -all now
mx seq=name iso=sample.iso -all now
```

Cuando MX realiza uno o más procesos que modifican registros (tanto si fueron leídos desde una base de datos, un archivo ISO-2709 o un archivo de texto), esas modificaciones se llevan a cabo en memoria y **no modifican la base de datos**, a menos que se indique explícitamente.

Los registros modificados pueden verse por pantalla o grabarse en una base de datos o en un archivo de salida.

Entre los principales procesos de modificación podemos nombrar:

Procedimientos para cambios globales de patrones (*gizmo*).

Unión de registros (o parte de estos) provenientes de otra base de datos (*join*).

Realizar las operaciones de actualización de campos (*proc*).

Ejecutar una Tabla de Selección de Campos de MicroISIS y agregar los datos del resultado al registro en memoria (*fst*).

El siguiente ejemplo muestra los registros del archivo maestro cds, que fueron elegidos por el usuario al indicar el número de registro a visualizar a través del teclado.

- MS-DOS:

```
mx seq=con "join=cds='mfn='v1" "proc='D1/1D32001'"
```

- UNIX:

```
mx seq=/dev/ttyp0 "join=cds='mfn='v1" "proc='D1/1D32001'"
```

MX puede actualizar los registros modificados sobre el mismo archivo maestro que se utilizó como entrada:

```
mx cds "proc='D24'" copy=cds -all now
```

El ejemplo borra el campo con etiqueta (*tag*) 24 de todos los registros de la base de datos cds, realizando los cambios sobre la misma base.

MX puede tomar los parámetros de un archivo de texto, permitiendo así superar las limitaciones del sistema operativo que se presentarían si:

1. Una llamada al MX tiene más caracteres que los que permite en una línea de comandos el sistema operativo (128 caracteres en MS-DOS y 512 caracteres en algunos UNIX).

2. La línea de comandos que se está ingresando por teclado contiene caracteres especiales para el sistema operativo.

El siguiente ejemplo muestra cómo utilizar un archivo de parámetros:

```
mx in=somefile
```

Donde el archivo *somefile* contiene:

```
\cisis\bases\cds
proc='D1'
copy=\cisis\bases\cds
-all
now
```

Sintaxis

MX versión 5.2a, tabla de sintaxis:

```
CISIS Interface v5.2a/PC32/L/M/32767/16/60/I - Utility MX
Copyright (c)BIREME/PAHO 2006. [http://www.bireme.br/products/cisis]

mx [cipar=<file>] [{mfrl|load}=<n>] [cgi={mx|<v2000_fmt>}] [in=<file>]
   {[db=]<db>|
    seq[/1m]=<file>|
    iso[={marc|<n>}]=<isofile> [isotag1=<tag>]|
    dict=<if>[,<keytag>[,<posttag>[/<postsperrec>]]] [k{1|2}=<key>]}

options:
   {from|to|loop|count|tell|btell}=<n>
   text[/show]=<text>
   [bool=]{<bool_expr>|@<file>}  [invx=<tag101_mf>] [tmpx=<tmp_mf>]

   gizmo=<gizmo_mf>[,<taglist>]  [gizp[/h]=<out_mfx>] [decod=<mf>]

   join=<mf>[:<offset>][,<taglist>]=<mfn=_fmt>
   join=<db>[:<offset>][,<taglist>]=<upkey_fmt> [jmax=<n>]
   jchk=<if>[+<stwfile>]=<upkey_fmt>

   proc=[<proc_fmt>|@<file>]
```

```
        D{<tag>[/<occ>]|*}
        A<tag><delim><data><delim>
        H<tag> <length> <data>
        <TAG[ <stripmarklen>[ <minlen>]]><data></TAG>

        S[<tag>]
        R<mf>,<mfn>
        G<gizmo_mf>[,<taglist>]
        Gsplit[/clean]=<tag>[={<char>|words|letters|numbers|trigrams}]
        Gsplit=<tag>=6words[/if=<if>]
        Gload[/<tag>][/nonl][=<file>]
        Gmark[/<tag>]{/<elem>|/keys|/decs|/<mf>,<otag>[,<ctag>]}=<if>
        Gmarx[/<tag>]/<elem>[@<att>="x"] =<tag>[:&[<att>]|/c[=224]|/i]
        Gdump[/<tag>][/nonl][/xml][=<file>]
        =<mfn>
        X[append=]<mf>

    convert=ansi [uctab={<file>|ansi}]          [actab={<file>|ansi}]
    fst[/h]={<fst>|@[<file>]}                   [stw=@[<file>]]

    [mono|mast|full] {create|copy|append|merge|updatf}=<out_mf>
    [out]iso[={marc|<n>}]=<out_isofile>         [outisotag1=<tag>]
    fullinv[/dict][/keep][/ansi]=<out_if>       [maxmfn=<n>|master=<mf>]
    ln{1|2}=<out_file>                          [+fix[/m]]
    fix=<out_file> tbin=<tag>
    tab[/lines:100000/width:100/tab:<tag>]=<tab_fmt>
    {prolog|pft|epilog}={<diplay_fmt>|@<file>} [lw={<n>|0}]

    {+|-}{control|leader|xref|dir|fields|all} mfrl now
```

MX toma los parámetros en el orden que muestra la tabla. En primer lugar deben ir, si los hubiera, los parámetros de inicialización (setup), luego la fuente de entrada de datos,y en último lugar deben presentarse los parámetros opcionales de proceso. Hay algunas excepciones que se indican en el manual, por ejemplo *btell=* debe ir antes que *bool=*.

Parámetros. Descripción general

Si se ingresa el nombre del programa MX sin parámetros, se desplegará el menú de todas las opciones posibles y una breve descripción de su uso, tal como se muestra en el cuadro anterior.

Parámetros de inicialización (setup)

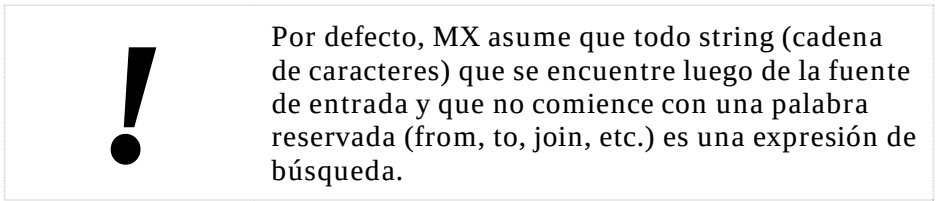
Parámetros opcionales de inicialización (files, mfrl, fmtl, load), cuando uno o más están presentes, deben colocarse antes que cualquier otro parámetro.

Parámetros que indican la fuente de entrada de datos

Un parámetro obligatorio que indica la fuente de entrada de datos (nombre de la base de datos, archivo ISO-2709 o archivo de texto), debe ser el primer parámetro, excepto que en la llamada existan parámetros de inicialización, en cuyo caso debe ubicarse inmediatamente después de estos.

Parámetros para procesamiento de datos

Parámetros opcionales que realizan tareas sobre los datos que ingresan. Siguen en la línea de comandos al parámetro que indica la fuente.



Por defecto, MX asume que todo string (cadena de caracteres) que se encuentre luego de la fuente de entrada y que no comience con una palabra reservada (from, to, join, etc.) es una expresión de búsqueda.

Los parámetros de proceso pueden clasificarse en:

Parámetros para selección de registros

Con estos parámetros se define un subconjunto de la entrada sobre el cual se trabajará. La forma de definir este subconjunto puede ser por:

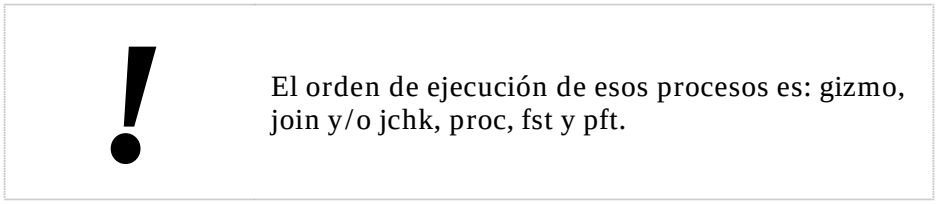
- Una búsqueda (*bool*)
- Un patrón con el que se realiza una búsqueda por texto libre (*text*)
- Un rango de registros (cuyos límites se indican con *from*, *to*)
- Cantidad de registros (*count*)
- Salto entre registro y registro a procesar (*loop*)

Parámetros que realizan procesos

Son parámetros que llaman a procedimientos internos que realizan tareas en memoria sobre el conjunto de registros leídos.

Estas tareas pueden ser:

- Realizar cambios globales (*gizmo*)
- Unir registros (*join*)
- Confrontar archivos maestros con archivos invertidos (*jchk*)
- Realizar modificaciones en los campos de los registros (*proc*)
- Aplicar Tablas de Selección de Campos (*fst*) a los registros
- Aplicar formatos a los registros (*pft*)



Parámetros de salida de datos

Son parámetros que permiten, por ejemplo, indicar:

- La base de datos de salida (*create*, *copy*, *append*, etc.)
- El nombre de un archivo ISO-2709 de salida (*iso*)
- El nombre de archivos de ligas (*ln1*, *ln2*)
- Llamadas al sistema operativo (sys)

Parámetros generales

Son parámetros que realizan tareas generales, por ejemplo:

Desactivar el *prompt* (interacción con el usuario) entre registro y registro procesados (*now*)

Cambiar el texto de los *prompts* del mx (*p1*, *p2*)

Modificar opciones de visualización (+*fields*, +*all*, -*all*, etc.).

- Redireccionar la salida estándar (>, >>)
- Seguimiento de la ejecución de CISIS (*trace=rec*, *trace=all*, etc.).

Parámetros que indican cuál es la fuente de entrada de datos

Base de datos de entrada

<[db=]<db>>

Especifica el archivo maestro a ser leído. Los procesos a realizarse se ejecutarán sobre los registros de este archivo maestro.

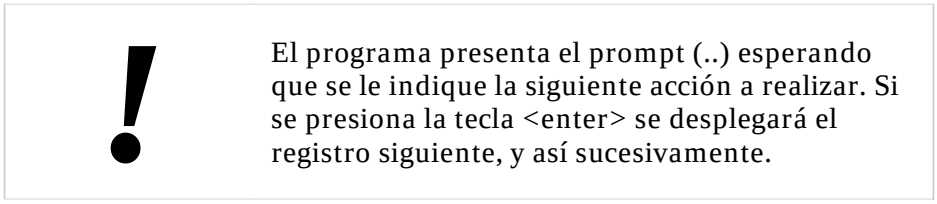
```
C:\isis\data> mx cds
```

o

```
X:\otrodirectorio> mx C:\isis\data\cds
```

Genera una salida sin formatear, mostrando todos los campos de a un registro por vez.

```
mfn= 1
44 «Methodology of plant eco-physiology: proceedings of the Montpellier Symposium»
50 «Incl. bibl.»
69 «Paper on: <plant physiology><plant transpiration> <measurement and instruments>»
24 «Techniques for the measurement of transpiration of individual plants»
26 «^aParis^bUnesco^c-1965»
30 «^ap. 211-224^billus.»
70 «Magalhaes, A.C.»
70 «Franco, C.M.»
..
```




```
mfn= 2

44 «Methodology of plant eco-physiology: proceedings of the Montpellier
Symposium»

50 «Incl. bibl.»

69 «Paper on: <plant evapotranspiration>» 24 «<The> Controlled climate in
the plant chamber and its influence upon assimilation and transpiration»

26 «^c1965»

30 «^ap. 225-232^billus.»

70 «Bosian, G.»

..
```

Frente al *prompt(..)* es posible tomar tres acciones:

a) Seguir desplegando registros presionando <enter>
b) Ingresar una *x* minúscula y presionar <enter> para salir del programa.
c) Cualquier otro dato que ingrese será interpretado como una expresión de búsqueda y luego de presionar <enter> MX procederá a realizar la búsqueda y a mostrar los registros recuperados.

Archivo ISO-2709 de entrada

iso[={marc|<n>}]=<isofile> [isotag1=<tag>]

En la sección anterior se consideraron como fuente de entrada a bases de datos en formato CDS/ISIS. El programa MX puede leer indistintamente archivos en formato ISO-2709 y aplicar los mismos procedimientos que sobre archivos maestros (exceptuando aquellos procesos que requieren el uso del archivo invertido o diccionario como, por ejemplo, una búsqueda).

Cada registro en formato ISO-2709 que ingrese es transformado internamente en un registro en formato ISIS, sobre el cual se trabaja.

Ejemplos:

- Recorrer un archivo ISO-2709:

  ```
  mx iso=\isis\sys\cds.iso
  ```

  Recorre un archivo con formato ISO-2709 llamado *cds.iso* que se encuentra en directorio *\isis\sys*.
- Leer un archivo ISO y crear un archivo maestro con el contenido de aquel. El *prompt* es suprimido por el parámetro *now* (no wait).

  ```
  mx iso=cds.iso create=newcds now
  ```

- Leer un archivo con formato ISO-2709 que se encuentra en el directorio corriente y crear un archivo maestro llamado *newcds* con los datos proporcionados por el archivo ISO.

  Este ejemplo realiza el mismo proceso que el anterior pero sin presentar información en pantalla. El parámetro *-all* desactiva el vuelco de información en pantalla.

  ```
  mx iso=cds.iso create=newcds now -all
  ```

- Leer un archivo ISO y crear un archivo maestro llamado *newcds* con los registros de entrada, que comienza en el mfn 1001. Al igual que el ejemplo anterior no se desplegará información en pantalla (*-all*) ni se detendrá entre registro y registro esperando la intervención del usuario (*now*).

  ```
  mx iso=cds.iso create=newcds from=1001 -all now
  ```

Largo de línea fijo

El MX puede leer archivos ISO con largo de línea fijo. Esta opción se utiliza para intercambiar archivos ISO de PCs con computadoras y software que así lo

requieran (como el MINIISIS de HP). Para leer archivos con largo de línea fijo, se debe indicar el largo de la línea a continuación del parámetro *ISO*:

```
mx iso=80=cds.iso create=newcds now -all
```

El ejemplo lee el archivo ISO suponiendo que el largo de línea es de 80 caracteres.

Si el largo de las líneas es variable, entonces el parámetro será “0” (cero).

```
mx iso=0=cds.iso create=newcds2 now -all
```

Lectura de archivos MARC

Los archivos que cumplen con la norma MARC registran como fin de cada línea solamente el carácter ‘\0Bx’ y no indican ‘\0Cx’ (<CR>) adicional. Para estos archivos se indicará el parámetro como sigue:

```
mx  iso=marc=input.iso  create=output.iso now –all
```

Datos del Leader del registro MARC

Los registros MARC contienen en el Leader datos que no se convierten automáticamente al CDS/ISIS. Si estos datos fueran necesarios deberán convertirse a campos convencionales indicando un valor en el parámetro isotag1=<n> como base a partir del cual se cargarán los bytes del leader. Por ejemplo, si se indica isotag1=3000, el byte posición 5 del leader se cargará en el campo 3005.

```
mx iso=marc=input.iso  isotag1=3000  create=newcds now -all
```

Archivo de texto ASCII de entrada

seq[/1m]=<file>

<file>={filename|con|null}

MX puede tomar como fuente de entrada un archivo de texto ASCII plano. El largo de la línea puede llegar hasta 1Mb, lo que se indica con /1m.

Cada línea de este archivo de entrada se convertirá en un registro del archivo maestro. Los datos dentro de cada línea podrán separarse con delimitadores, de esta manera cada parte de la línea se ingresa en el registro como campos consecutivos. La cantidad de campos será uno más que la cantidad de delimitadores.

El delimitador predefinido en MX es la barra vertical (**|** *pipe*). Cada línea del archivo ASCII de entrada puede tener hasta 32743 caracteres de largo para la carga de un campo, 32740 para dos campos, 32735 para tres campos, etc.

Si se tiene un archivo de texto con una sola línea, llamado input.in, cuyo contenido es:

```
agua|tierra|abono
```

Con la línea:

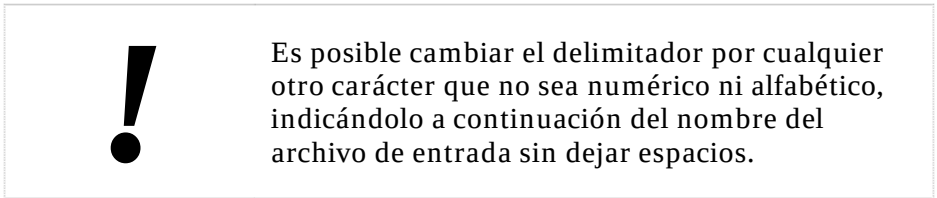
```
mx seq=input.in create=salida now
```

Se convertirá en:

```
mfn= 1
1 «agua»
2 «tierra»
3 «abono»
```

Dos delimitadores consecutivos producirán un salto en el número de campos:

```
agua||tierra||abono
mfn= 1
1 «agua»
3 «tierra»
5 «abono»
```



! Es posible cambiar el delimitador por cualquier otro carácter que no sea numérico ni alfabético, indicándolo a continuación del nombre del archivo de entrada sin dejar espacios.

En el ejemplo siguiente el punto y coma (**;**) se usará como delimitador.

El archivo input.in contiene la línea:

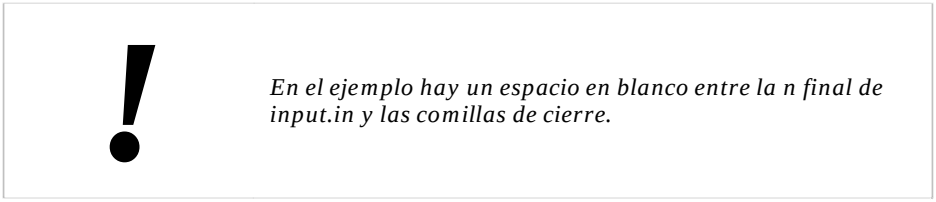
```
agua;abono;tierra
```

Para leer el archivo input.in con punto y coma (;) como separador de campos tipear:

```
mx "seq=input.in;" create=salida now
```

Es posible también usar como delimitador al espacio en blanco:

```
mx "seq=input.in " create=salida now
```



En el ejemplo hay un espacio en blanco entre la n final de input.in y las comillas de cierre.

El archivo de entrada puede generarse ingresando los datos directamente desde el teclado, esto es, utilizando el dispositivo estándar de entrada *con* (consola) como fuente de entrada datos. Por consiguiente, es posible crear registros en una base CDS/ISIS escribiendo los datos directamente desde el teclado.

Examínese la siguiente secuencia:

PROCESO DEL PARÁMETRO *con*		
Pasos	**Explicación**	**Líneas a tipear**

1	Desde el *prompt* del sistema operativo se da la instrucción	`C:\path> mx seq=con create=out`
		`(MS-DOS)`
		`unixuser:~$ mx seq=/dev/tty1 create=out`
		`(entrada desde la consola 1 en UNIX)`

2	Se inicia una sesión de ingreso de línea que termina al presionar <enter>	`agua|tierra|vegetales|abono<enter>`

3	Automáticamente se crea el registro mfn=1 en el archivo maestro *out*
	y se presentan los campos del registro
		`mfn= 1`
		`1 «agua»`
		`2 «tierra»`
		`3 «vegetales»`
		`4 «abono»`

PROCESO DEL PARÁMETRO *con*		
Pasos	**Explicación**	**Líneas a tipear**
4	El *prompt* del MX queda esperando nuevas instrucciones. Se continua la creación de registros ingresando <enter> a la solicitud del *prompt* del MX	..<enter>
5	Se ingresa una nueva línea que se termina al presionar <enter>	bovinos|ovinos|equinos<enter>
6	Se crea el registro mfn=2 y se presentan los campos del registro	mfn= 2 1 «bovinos» 2 «ovinos» 3 «equinos»
7	Se termina la creación de registros ingresando una *x* minúscula a la solicitud del *prompt* del MX	..x<enter>
8	Finaliza la ejecución del comando	C:\path> (MS-DOS) unixuser:~$ (UNIX)

El proceso puede agilizarse evitando la solicitud del *prompt* del MX poniendo el parámetro *now*. En este caso para terminar el proceso de ingreso deberá presionarse las teclas <ctrl>+<Z> o <F6> (en MS-DOS) y <ctrl>+<D> (en UNIX).

La siguiente tabla describe el proceso utilizando el parámetro *now*:

PROCESO DEL PARÁMETRO *now*		
Pasos	**Explicación**	**Líneas a tipear**
1	Desde el *prompt* del sistema operativo se da la instrucción	C:\path>mx seq=con create=out now (MS-DOS) unixuser:~$ mx seq=/dev/tty1 create=out now (entrada desde la consola 1 en UNIX)
2	Se inicia una sesión de ingreso de línea que termina al presionar <enter>	agua|tierra|vegetales|abono<enter>

PROCESO DEL PARÁMETRO *now*		
Pasos	**Explicación**	**Líneas a tipear**

3	Se crea el registro mfn=1
	y se presentan los campos del registro
		`mfn= 1`
		`1 «agua»`
		`2 «tierra»`
		`3 «vegetales»`
		`4 «abono»`

4	Se ingresa una nueva línea que se termina al presionar <enter>	`bovinos|ovinos|equinos<enter>`

5	Se crea el registro mfn=2
	y se presentan los campos del registro
		`mfn= 2`
		`1 «bovinos»`
		`2 «ovinos»`
		`3 «equinos»`

6	Se finaliza la creación de registros directamente desde el teclado ingresando las teclas <ctrl>+<Z> (o la tecla F6) en DOS, o las teclas <ctrl>+<D> en UNIX	`<Ctrl>+<Z> o <F6>`
		`(MS-DOS)`
		`<ctrl>+<D>`
		`(UNIX)`

7	Finaliza la ejecución del comando	`C:\path>`
		`(MS-DOS)`
		`unixuser:~$`
		`(UNIX)`

Base de datos ficticia

con | null

MX permite que la fuente de entrada sea una base de datos ficticia (*dummy*) llamada *null* (indistintamente). La base *null* consiste de un número ilimitado de registros activos sin campos de datos (registros vacíos). Es posible ingresar directamente los datos desde el teclado asignando el parámetro de entrada a *con* (consola). Cada <enter> creará un registro. Para terminar el proceso deberá darse <ctrl>+Z.

La utilización de este parámetro no está relacionado directamente con las operaciones que se realizan sobre una base de datos. Se utiliza en procesos que involucran contar, generar listas de números, etc.

Ejemplos:

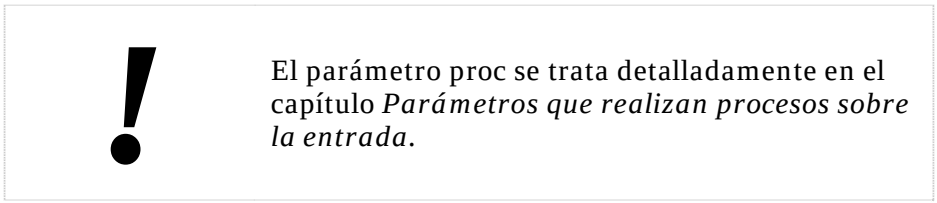
- Desplegar una lista de números consecutivos desde 1 a 50:

```
mx null to=50 pft=mfn(3)/ -all now
```

- Crear un archivo maestro OUT con registros del 100 al 200 conteniendo el literal DBN en el campo 999:

```
mx null from=100 to=200 proc='A999#DBN#' now -all create=OUT
```

El parámetro *proc* realiza el alta y la baja de campos en un registro. En el ejemplo, proc agrega el campo 999 (*A999*) con contenido DBN (*#DBN#*).



El parámetro proc se trata detalladamente en el capítulo *Parámetros que realizan procesos sobre la entrada*.

Archivo de parámetros

in=<file>

Los parámetros del programa MX pueden ser leídos desde un archivo ASCII externo, donde cada parámetro se ingresa como una línea separada en ese archivo. De esta manera es posible programar líneas de comandos para MX que se extiendan más allá de lo admitido por el sistema operativo (normalmente limitado en 128 caracteres de longitud para MS-DOS y en 512 caracteres para UNIX).

Suponiendo que se tiene un archivo llamado input.in, cuyo contenido es:

```
iso=entrada.iso
create=salida
now
from=10
to=20
-all
```

La línea de comando:

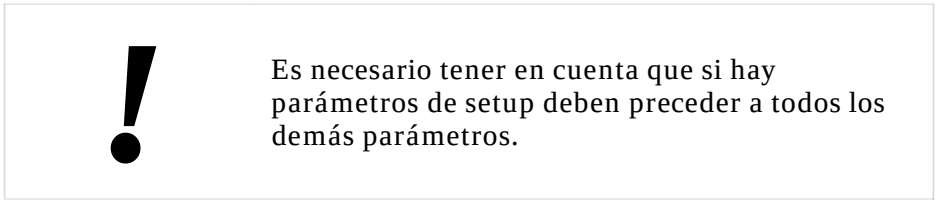
```
mx iso=entrada.iso create=salida now from=10 to=20 -all
```

Podría escribirse como:

```
mx in=input.in
```

El parámetro *in* puede usarse en cualquier parte de la línea de comandos a partir de la primera posición, puede ser el único parámetro o puede ser sólo una parte de la línea de parámetros del MX.

Si se encuentra en la primera posición (o es el único parámetro) la primera línea de ese archivo debe ser el nombre del archivo de datos de entrada.



Es posible usar más de un archivo *in* en una línea de comando, y usar el parámetro *in* dentro de un archivo *in* hasta en 8 niveles de recursividad.

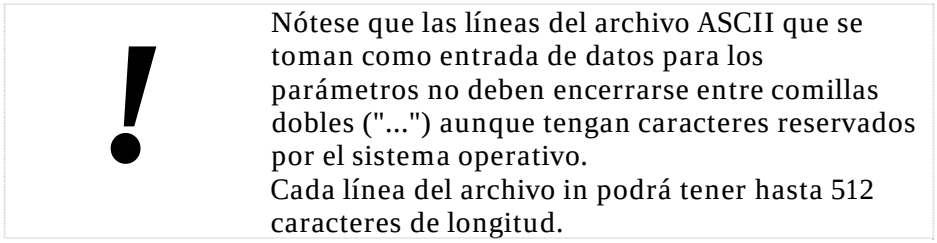
El siguiente ejemplo presenta una llamada al MX utilizando un archivo de parámetros *in* entre otros parámetros:

Sea el archivo print:

```
\cisis\bases\cds
pft=v70+|; |, " / "v26". "/# mhl,(v69/)
now
-all
tell=10
```

La línea de comando podría ser:

```
mx in=print from=10 to=50 > archivo.txt
```



El uso del parámetro *in* permite preparar archivos de tareas por lotes (*bat* de MS-DOS y *scripts* de UNIX). Sea el archivo imprimir.bat cuyo contenido es el siguiente:

Versión MS-DOS

```
REM Debe ingresar los valores from= to=
REM Por ejemplo: print 10 20
mx in=print from=%1 to=%2 > archivo.txt
```

Versión UNIX

```
# Debe ingresar los valores from= to=
# Por ejemplo: print 10 20
mx in=print from=$1 to=$2 > archivo.txt
```

Recuerde que el archivo imprimir.bat en UNIX requiere permiso de ejecución.

El operador podría indicar por ejemplo:

- MS-DOS

```
C:\cisis\sys> imprimir 10 40
```

- UNIX

```
unixuser:~/cisis/sys$ imprimir 10 40
```

Lo que se traducirá a:

```
mx in=print from=10 to=40 > archivo.txt
```

Esto a su vez se convertirá en:

```
mx \cisis\bases\cds pft=v70+|; |, " / "v26". "/# mhl,(v69/) now -all
tell=10 from=10 to=40 > archivo.txt
```

Archivo Invertido como entrada

dict=<if>[,<keytag>[,<posttag>[/<postsperrec>]]]
[k{1|2}=<key>]

MX lee como entrada de datos las claves de un archivo invertido produciendo registros con etiqueta de campo <ktag>, entre las claves k1 y k2, con los siguientes parámetros.

If	Nombre del archivo invertido (I/F)
keytag	Etiqueta que tendrá la clave leída (*key tag*), por defecto 1 Este campo tiene los siguientes subcampos ^l el tipo de clave (1=corta, 2=larga) ^s el tamaño efectivo de la clave (cantidad de caracteres = *keylength*) ^t total de apuntadores (*totalpostings*) ^k número secuencial de lectura (*keyorder*)
posttag	Etiqueta que tendrá el apuntador leído (*posting tag*) Si se indica este parámetro, MX combinará las claves del I/F y sus postings correspondientes. El campo posttag tendrá los siguientes subcampos ^m MFN del apuntador exhibido ^t la etiqueta de campo donde fue localizada la clave ^o la ocurrencia de la etiqueta en que fue localizada la clave ^c la posición, contada en palabras, de la clave en ese campo ^p el número secuencial del apuntador en exhibición ^k el número secuencial de lectura de la clave
postsperrec	Máximo de apuntadores leídos por registro, por defecto todos (*all*)
k1=<key>	Clave inicial a exhibir
k2=<key>	Clave final a exhibir

Ejemplo 1 Lee las claves del archivo invertido CDS comprendidas en el intervalo entre los términos AFRICA y AFRICAN LANGUAGES

```
Mx dict=cds   k1=africa  "k2=african languages"  now
mfn= 1
1 «AFRICA^l1^s6^t5^k1»
mfn= 2
1 «AFRICAN LANGUAGES^l2^s17^t1^k2»
```

Ejemplo 2 Lee tres claves y postings del archivo invertido CDS desde el término AFRICA

```
mx dict=cds,1,2 k1=africa count=3 now
mfn=     1
  1  «AFRICA^l1^s6^t5^k1»
  2  «^m93^t69^o1^c9^p1^k1»
  2  «^m111^t24^o1^c3^p2^k1»
  2  «^m111^t69^o1^c3^p3^k1»
  2  «^m115^t69^o1^c7^p4^k1»
  2  «^m121^t69^o1^c4^p5^k1»
mfn=     2
  1  «AFRICAN LANGUAGES^l2^s17^t1^k2»
  2  «^m75^t69^o1^c10^p1^k2»
mfn=     3
  1  «AGE^l1^s3^t1^k3»
  2  «^m136^t24^o1^c7^p1^k3»
```

Ejemplo 3 Lee las claves del archivo invertido CDS exhibiendo como máximo tres postings cada vez desde el término AFRICA, muestra tres registros de salida, en este caso tres claves

```
mx dict=cds,1,2/3   k1=africa   count=3 now
mfn=     1
  1  «AFRICA^l1^s6^t5^k1»
  2  «^m93^t69^o1^c9^p1^k1»
  2  «^m111^t24^o1^c3^p2^k1»
mfn=     2
  1  «AFRICA^l1^s6^t5^k1»
  2  «^m115^t69^o1^c7^p4^k1»
  2  «^m121^t69^o1^c4^p5^k1»
mfn=     3
  1  «AFRICAN LANGUAGES^l2^s17^t1^k2»
  2  «^m75^t69^o1^c10^p1^k2»
```

Ejemplo 4 Lee las claves del achivo invertido CDS exhibiendo los postings del término AFRICA uno por uno

```
mx dict=cds,1,2/1   k1=africa   count=5 now
mfn=     1
  1  «AFRICA^l1^s6^t5^k1»
  2  «^m93^t69^o1^c9^p1^k1»
mfn=     2
  1  «AFRICA^l1^s6^t5^k1»
  2  «^m111^t24^o1^c3^p2^k1»
mfn=     3
  1  «AFRICA^l1^s6^t5^k1»
  2  «^m111^t69^o1^c3^p3^k1»
mfn=     4
  1  «AFRICA^l1^s6^t5^k1»
  2  «^m115^t69^o1^c7^p4^k1»
mfn=     5
  1  «AFRICA^l1^s6^t5^k1»
  2  «^m121^t69^o1^c4^p5^k1»
```

Parámetros que realizan procesos sobre la entrada

Parámetros que aplican formatos a la entrada

Este parámetro suministra las especificaciones de formato para la visualización de los registros. Los registros borrados (*logically deleted*) no se visualizan a través del parámetro *pft=*.

El MX soporta todas las instrucciones de lenguaje de formateo de CDS/ISIS estándar para DOS (excepto *format exits*) y agrega algunas extensiones desarrolladas por la Interfaz CISIS. Muchas de estas nuevas instrucciones están incorporadas en Winisis, pero MX no acetpa las instrucciones para formatos gráficos en RTF que usa el Winisis.

El manual completo de formatos está disponible en el sitio del Modelo de la BVS, a través de la URL: http://bvsmodelo.bvsalud.org/.

Especificación del formato de visualización en la línea de comando

pft=<display_fmt>

El siguiente ejemplo aplica el formato mfn/v24/v26 sobre los registros obtenidos del archivo maestro de entrada (cds):

```
mx cds pft=mfn/v24/v26
```

Si la instrucción de formato incorpora caracteres reservados por el sistema operativo (tales como: > | % etc.) o espacios en blanco, el parámetro deberá encerrarse entre comillas dobles:

```
mx cds "pft=mfn,/(v70+|; |)/v24/#"
```

Especificación del formato de visualización mediante un archivo

pft= @[<file>]

El MX permite especificar un archivo (*pft=@[<file>]*) donde reside el formato a utilizar. Ésta es una manera más práctica de especificar un formato de visualización, así la llamada al MX es más clara y, por otro lado, no se pierde el formato una vez ejecutado el comando.

Además, de esta forma, se supera una limitación de los sistemas operativos, ya que la longitud de una línea de comandos es limitada (128 caracteres MS-DOS y 512 caracteres UNIX) por lo que un formato extenso no podría escribirse explícitamente.

Si no se provee nombre de archivo, entonces el MX usará por defecto el formato que tiene el mismo nombre de la base de datos:

mx cds pft=@ es equivalente a mx cds pft=@cds.pft

Al especificar un archivo, su nombre puede tener más de seis caracteres de longitud, puede estar ubicado en un directorio diferente al de la base de datos, y puede tener o no extensión (en el caso que la tuviere deberá escribirse, aún si ésta fuera pft).

Ejemplos:

```
mx cds pft=@cdsnew.pft
mx cds pft=@\dbisis\otro_dir\otro.pft
mx cds pft=@long_name.pft
mx cds pft=@sin_ext
```

Los registros borrados no se visualizan.

Formatos condicionales

prolog | epilog

prolog es una especificación que se cumple en el primer registro de ejecución de la salida, y *epilog* en el último registro.

Ejemplo:

```
mx  cds  prolog='Primero: '   pft=mfn/   epilog='ultimo'  from=10 to=20 now
```

Ancho de línea (*line width*)

lw={<n>|0}

La línea de salida tiene un ancho predefinido de 78 caracteres. Es posible cambiar el ancho de línea con el parámetro *lw=n*.

```
mx cds "pft=mfn,/(v70+|; |)/v24/#" lw=40 to=20 now
```

Extrae datos del contenido de una variable CGI

getenv('cgi=',<varfmt>)

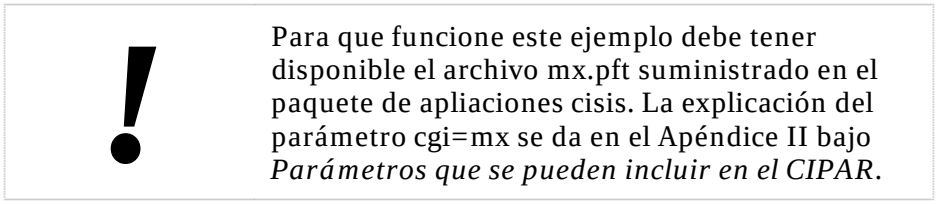
<varfmt> es una especificación de formato que genera un nombre de variable cgi.

Se separan las múltiples ocurrencias con el carácter %

```
set "REQUEST_METHOD=GET"
set "QUERY_STRING=db~cds&btell~0&bool~plants*water"
mx cds cgi=mx "pft='Buscando por \"',getenv('cgi=bool'),'\" en la base:
',getenv('cgi=db')/,'  MFN   Titulo'/,mfn,x2,mhl,v24.50,'...'/"
```

que despliega en la salida:

```
 Buscando por "plants*water" en la base: cds
  MFN   Titulo
000004  Mc An Electric hygrometer apparatus for me...
..
Buscando por "plants*water" en la base: cds
  MFN   Titulo
000011  Measurement of water stress in plants...
..
Buscando por "plants*water" en la base: cds
  MFN   Titulo
000013  Experience with three vapour methods for measuring...
->x
```



Para que funcione este ejemplo debe tener disponible el archivo mx.pft suministrado en el paquete de apliaciones cisis. La explicación del parámetro cgi=mx se da en el Apéndice II bajo *Parámetros que se pueden incluir en el CIPAR*.

Obtiene un archivo temporario vacío

getenv('tmp=',[<pathfmt>])

El formato obtiene un nombre de archivo temporario vacío en el directorio de trabajo actual, o en el que se especifica en la ruta (*path*) que se especifica en el formato <pathfmt>. Si se especifica 'ci_tmpdir' entonces la ruta se obtiene de las siguientes variables de entorno: *ci_tempdir*, *temp*, *tmp*.

```
mx null pft=getenv('tmp=')
TMP1.$$$

mx null pft=getenv('tmp=','ci_tempdir')
C:\windows\TEMP\TMP1.$$$

set ci_tempdir=C:\work
mx null pft=getenv('tmp=','ci_tempdir')
C:\work\TMP1.$$$

echo ci_tempdir=C:\work2 >xcip
mx cipar=xcip null pft=getenv('tmp=','ci_tempdir')
C:\work2\TMP1.$$$
```

Parámetros que seleccionan el conjunto de registros a ser procesados

Especificación de la expresión de búsqueda en la línea de comando

bool=<bool_expr_spec>

El parámetro *bool* permite realizar búsquedas en bases de datos. MX provee todas las operaciones booleanas del lenguaje de búsqueda de CDS/ISIS.

El resultado de una búsqueda es un conjunto de registros que cumplen con lo especificado en la expresión de búsqueda.

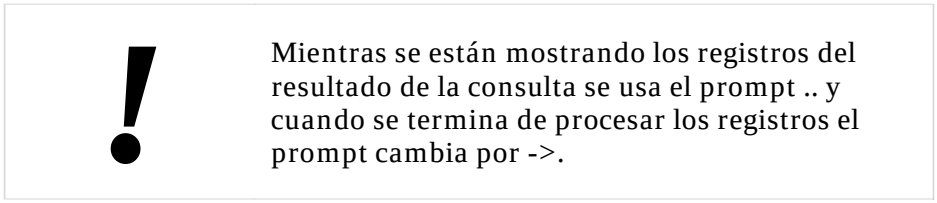
La palabra *bool* es opcional, ya que MX considera expresión de búsqueda todo aquello que no sea un parámetro. Los dos ejemplos siguientes producen el mismo resultado:

```
mx cds bool=water
mx cds water
```

La salida de la ejecución de la búsqueda es:

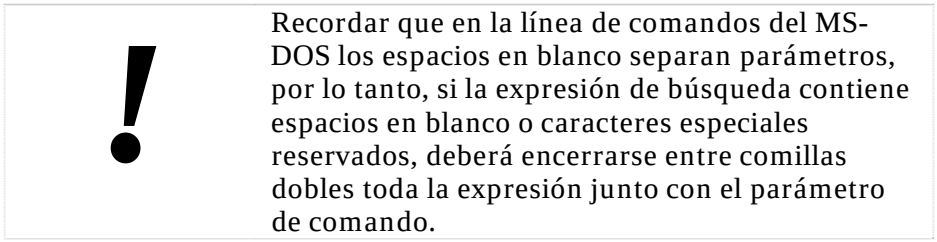
```
mx cds culture pft=mfn/
2       2      CULTURE
2       2      Set #000001
Hits=2
000050
..
000064
->
```

Primero se presenta la resolución de la búsqueda y se asigna al resultado un número de secuencia (*Set #000001*). Luego se despliegan los registros de acuerdo al formato prestablecido.



Mientras se están mostrando los registros del resultado de la consulta se usa el prompt .. y cuando se termina de procesar los registros el prompt cambia por ->.

Es posible seguir haciendo consultas desde el *prompt* en forma interactiva, pero no es posible, con este parámetro, referenciar a los resultados anteriores en consultas del tipo #1 + #2, etc. Para ello es necesario el uso del parámetro tmpx (se presenta más adelante en este capítulo).



Recordar que en la línea de comandos del MS-DOS los espacios en blanco separan parámetros, por lo tanto, si la expresión de búsqueda contiene espacios en blanco o caracteres especiales reservados, deberá encerrarse entre comillas dobles toda la expresión junto con el parámetro de comando.

```
mx cds "bool=water + soil"
mx cds "water + soil"
```

El programa MX acepta los operadores booleanos +, *, ^ que usa CDS/ISIS así como sus expresiones literales OR, AND, NOT.

```
mx cds "bool=water or soil"
mx cds "agricult$ and plants"
```

Cargar la expresión de búsqueda desde un archivo

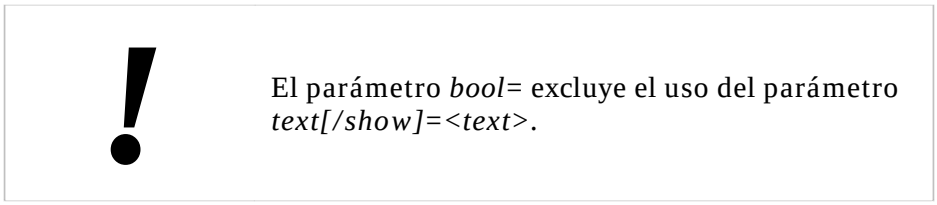
bool=@<file>

bool=@<*file*> toma la expresión de búsqueda desde un archivo ASCII externo. En este caso la expresión no requiere estar encerrada entre comillas.

Suponiendo que el archivo *consulta.001* contiene la expresión: agricult$ and plants, entonces el último ejemplo se traduce a cualquiera de las dos formas siguientes:

```
mx cds bool=@consulta.001
mx cds @consulta.001
```

El parámetro *bool* es optativo, pero podría ser necesario su uso cuando la expresión de búsqueda comienza con una palabra reservada del MX.



El parámetro *bool*= excluye el uso del parámetro *text[/show]=<text>*.

Si se quieren recuperar registros que contengan la palabra now:

`mx cds now`	No hará lo esperado, ya que interpreta a now como parámetro, no como expresión de búsqueda.
`mx cds "now"`	No hará lo esperado, porque interpreta a now como parámetro y no como expresión de búsqueda.
`mx cds bool=now`	Es correcto.
`mx cds "bool=now"`	Es correcto.

Utilización de los resultados intermedios de una búsqueda

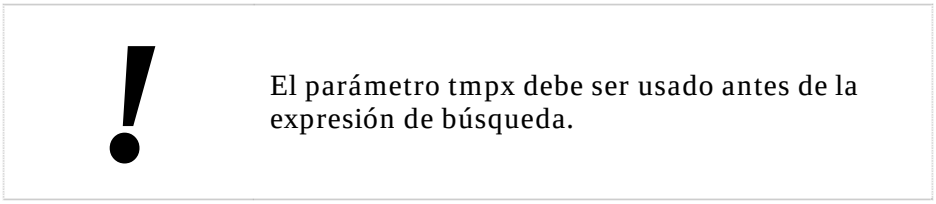
tmpx=<tmpx_query_dbn>

parámetro obsoleto:

b70=<b70_query_dbn>

El parámetro *tmpx* almacena los resultados intermedios de una sesión de búsquedas, permitiendo referenciar resultados anteriores en formulaciones del tipo #1 + #2.

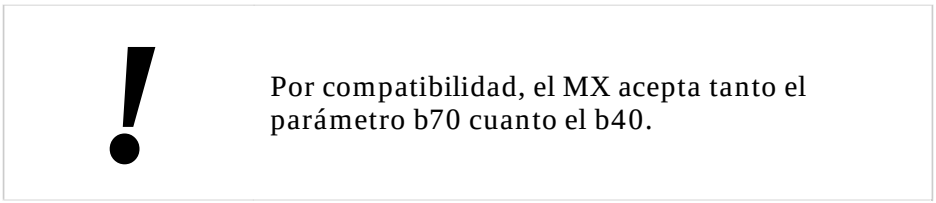
El nombre asignado a la derecha del signo de igual es una base de datos ISIS que MX crea y reinicializa automáticamente, y no es borrada al final de la ejecución.



El parámetro tmpx debe ser usado antes de la expresión de búsqueda.

```
mx cds tmpx=x70 plants water #1*#2 pft=mfn/ now
      8  PLANTS
      8  Set #000000001
Hits=8
     14  WATER
     14  Set #000000002
Hits=14
      3  Operation *
      3  Set #000000003
Hits=3
000004
000011
000013
```

Nótese que si se indica el parámetro *tmpx* es posible realizar consultas interactivas desde el *prompt* de MX haciendo referencias a resultados anteriores.



Por compatibilidad, el MX acepta tanto el parámetro b70 cuanto el b40.

En el siguiente ejemplo los términos que son introducidos uno a uno por el operador, como respuesta al *prompt*, se presentan resaltados en negrita. El resultado del proceso se almacena en la base X70.

```
mx cds tmpx=x70 plants pft=mfn/
      8  PLANTS
      8  Set #000000004
Hits=8
000001
..water
     14  WATER
```

```
     14  Set #000000005
Hits=14
000004
..#1 and #2
      3  Operation *
      3  Set #000000006
Hits=3
000004
..
000011
..
000013
->x
```

Es posible realizar consultas complejas leyendo la expresión a buscar desde un archivo externo. Suponiendo que el archivo PERFIL.001 tiene la siguiente formulación de búsqueda:

```
Water
Plants
#1 and #2
Transpiration
#3 or #4
```

La siguiente línea de comando produce una salida a archivo de texto con los registros recuperados de acuerdo al perfil señalado. Obsérvese que cada consulta individual debe ingresarse como líneas independientes en el archivo PERFIL.001.

```
mx CDS tmpx=x70 in=perfil.001 pft=@cds.pft now > out_001.txt
```

Eliminación de la estadística de la búsqueda

btell=0

Es posible suprimir la salida en pantalla que registra el proceso de los pasos intermedios y final de la búsqueda. En este caso el parámetro *btell=0* debe ir antes de la formulación de búsqueda.

Ejemplo:

Sin parámetro *btell=*	• **Con parámetro *btell=0***
`mx cds "water * plants" pft=mfn/ now`	`mx cds btell=0 "water * plants" pft=mfn/ now`

`14 WATER`
`8 PLANTS`
`3 Operation *`
`3 Set #000000001`
`Hits=3`
`000004`
`000011`
`000013`
	`000004`
	`000011`
	`000013`

btell=2

Lista los términos del diccionario cuando opera el $ de truncado.

Búsqueda en varios archivos invertidos

invx=M/F

El invx contiene registros con el campo v101 repetible, con el siguiente formato:

```
^pPrefijo o asterisco^yArchivo[^uUseprefix][^mMessage]
```

Ejemplo:

Se crea un master **cdsinvx**, con los siguientes campos

```
^p*^ycds^mText words
^pAU ^ycdsaut^mAuthor
^pTI ^ycdstit^mTitle words
^pKW ^ycdkw^mKeywords
```

Se crean los invertidos asociados a CDS

```
mx cds "fst=1 0 (v70/)" fullinv=cdsaut
mx cds "fst=1 4 v24" fullinv=cdstit
mx cds "fst=1 2 v69" fullinv=cdskw
```

Consultas

```
mx cds invx=cdsinvx "KW plants " pft=mfn,x1,v24,x1,v69
     2 PLANTS
```

```
      2  Set #000000001
Hits=2
000054 Vegetation as a geological agent in tropical deltas Paper on:
<vegetation><geology><deltas><tropical zones><sedimentation><plants><organic
matter><wetlands>..
000069 Some important animal pests and parasites of East Pakistan Paper on:
<pests><parasites><biology><ecology><plants><agriculture><public
health><food><Bangladesh>
->x
```

```
mx cds invx=cdsinvx "KW plants * east"
      2  PLANTS
      9  EAST
      1  Operation *
      1  Set #000000001
Hits=1
mfn=    69
 24  «Some important animal pests and parasites of East Pakistan»
 26  «^c1966»
 30  «^ap. 285-291^billus.»
 44  «Scientific problems of the humid tropical zone deltas and their implicatio
ns: proceedings of the Dacca Symposium»
 50  «Incl. bibl.»
 69  «Paper on: <pests><parasites><biology><ecology><plants><agriculture><public
 health><food><Bangladesh>»
 70  «Yosufzai, H.K.»
100  «1001»
->x
```

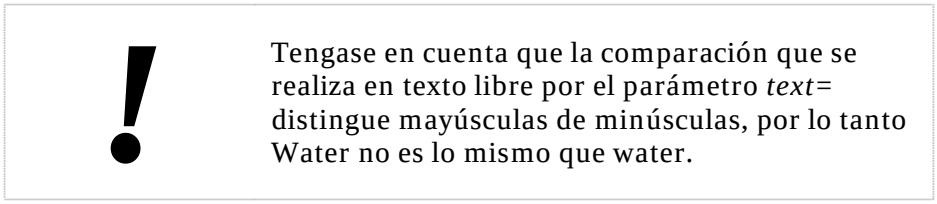
Si el prefijo se indica entre [ ], entonces se aplica a todos los términos de la formulación booleana:

```
mx cds invx=cdsinvx "[KW] plants * east"
      2  PLANTS
         EAST
         Operation *
         Set #000000001
Hits=0
->x
```

Búsquedas en texto libre

text[/show]=<text>

Esta función permite realizar búsquedas en texto libre (no indizado). Busca la cadena de caracteres indicada en <*text*> en todos los campos de la base de datos de entrada y selecciona los registros que cumplen con la búsqueda.



Ejemplo:

```
mx cds text=water iso=salida.iso now -all
```

El ejemplo realiza una búsqueda y exporta a un archivo ISO-2709 los registros encontrados.

Parámetro text/show

text/show

El parámetro *text/show* sólo despliega la información del registro y campo donde se produce la recuperación. Despliega dos líneas, una que contiene número de registro, etiqueta, ocurrencia donde aparece el texto buscado y el texto buscado; la segunda línea presenta la ocurrencia completa donde se presenta el texto buscado.

Ejemplo:

```
mx cds text/show=water now
```

Salida en pantalla:

```
mfn 56|tag 69|occ 1|water
69 «Paper on: <regime of waters><sediment transport><deltas>»
```

Indica que en el registro mfn=56 contiene la cadena de caracteres water en la primera ocurrencia del campo con tag 69.

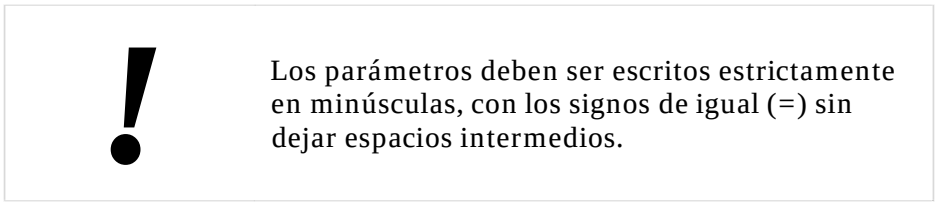
Otras formas de seleccionar el conjunto de registros a procesar

Selección por rango

from= <n> to=<n>

```
mx cds from=24 to=50
```

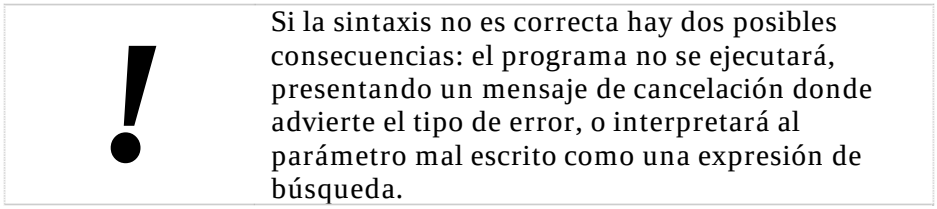
Esta línea de comandos despliega por pantalla desde el registro 24 hasta el registro 50 inclusive. Si no se indica from se asumirá, como comienzo, al primer registro de la base de datos; si no se indica to se procesará hasta el final de la base de datos, o hasta que el operador salga, presionando x.



Los parámetros deben ser escritos estrictamente en minúsculas, con los signos de igual (=) sin dejar espacios intermedios.

Los ejemplos siguientes NO son correctos:

`mx cds FROM=10`	El parámetro debe ir en minúsculas
`mx cds from= 10`	Hay un espacio en blanco entre el = y el 10
`mx cds from 10`	Falta el signo =



Si la sintaxis no es correcta hay dos posibles consecuencias: el programa no se ejecutará, presentando un mensaje de cancelación donde advierte el tipo de error, o interpretará al parámetro mal escrito como una expresión de búsqueda.

- La expresión del ejemplo (a) generará el resultado:

```
FROM=10
Set #000001
Hits=0
->
```

Esto significa que buscó la expresión FROM=10 en la base de datos.

- La expresión del ejemplo (b) generará el resultado:

```
FROM
Set #000001
Hits=0
Expression syntax error 5: '='
->
```

El MX interpreta que hay un error en la expresión de búsqueda debido al signo =

- La expresión del ejemplo (c) generará el resultado:

```
FROM
Set #00001
Hits=0
10
Set #000002
Hits=0
->
```

Esto implica que buscó las palabras FROM y 10 y no encontró ningún registro.

Con esta instrucción sólo se visualizarán los números de MFN de los registros 24 a 50.

```
mx cds from=24 to=50 pft=mfn/
```

Selección cada *n* registros

loop=<n>

El parámetro *loop=n* procesa un registro y saltea *n* registros, se procesa el *n+1*, saltea otros *n* y así sucesivamente.

Ejemplo:

```
mx cds loop=10
```

Se recuperan los registros 1, 11, 21, etc.

Seleccionar *n* registros

count=<n>

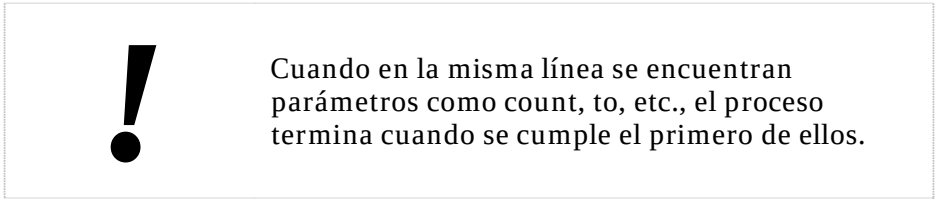
El parámetro *count=n* selecciona exactamente *n* registros a partir de un inicio dado. Si no se indica registro de inicio comienza desde el primer registro.

Ejemplo	**Salida**

`mx cds from=24 to=50 loop=5 pft=mfn/ now`	`000024`
	`000029`
	`000034`
	`000039`
	`000044`
	`000049`

`mx cds from=24 to=50 count=3 loop=5 pft=mfn/ now`	`000024`
	`000029`
	`000034`

`mx cds from=24 to=50 count=9 loop=5 pft=mfn/ now`	`000024`
	`000029`
	`000034`
	`000039`
	`000049`



Cuando en la misma línea se encuentran parámetros como count, to, etc., el proceso termina cuando se cumple el primero de ellos.

Parámetros que modifican registros

proc=[<proc_fmt>][@<proc_fmt_file>]

<proc_fmt> y <proc_fmt_file> sintaxis

```
D{<tag>[/<occ>]|*}
A<tag><delim><data><delim>
```

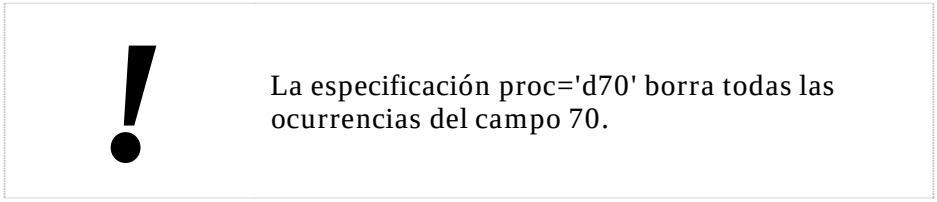
```
H<tag> <length> <data>
<TAG[ <stripmarklen>[ <minlen>]]><data></TAG>
S[<tag>]
R<mf>,<mfn>
G<gizmo_mf>[,<taglist>]
Gsplit[/clean]=<tag>[={<char>|words|letters|numbers|trigrams}]
Gload[/<tag>][/nonl][=<file>]
Gmark[/<tag>]{/<elem>|/keys|/decs|/<mf>,<otag>[,<ctag>]}=<if>
Gmarx[/<tag>]/<elem>[@<att>="x"] =<tag>[:&[<att>]|/c[=224]|/i]
Gdump[/<tag>][/nonl][/xml][=<file>]
=<mfn>
X[append=]<mf>
```

El parámetro *proc* permite especificar, mediante un formato, modificaciones a realizar sobre los campos del registro fuente. Posibilita borrar, agregar y reemplazar el contenido de los campos. Las operaciones a realizar se definen como instrucciones de formato, éste puede ser suministrado directamente en la línea de comandos o tomarse desde un archivo externo.

Por otra parte es importante destacar que las modificaciones de los registros no se realizan sobre la base origen (base de donde provienen los datos) sino sobre la base destino. En caso que la especificación de la base destino no se encuentre, los cambios se podrán ver en pantalla pero se perderán una vez terminada la ejecución. Para que los cambios se realicen sobre la misma base que se utilizó como entrada, ésta debe especificarse como base destino.

Ejemplos:

```
mx cds from=1 to=10 now proc='d70'
```



Los cambios se verán en pantalla pero no se realizarán realmente, porque no se especificó la base de destino. Para que los cambios se reflejen en el mismo archivo maestro que se especificó como entrada, hay que indicarlo mediante el parámetro *copy*:

```
mx cds from=1 to=10 proc='d70' copy=cds now
```

Otra posibilidad es especificar un archivo en lugar de escribir el formato en la línea de comandos:

```
mx cds from=1 to=10 now proc=@modifica
```

Realiza las modificaciones señaladas en un archivo que contiene el siguiente formato:

```
'd70'
```

Cantidad máxima de parámetros aceptados

in=	1024
archivos in= (recursivos)	16
proc=	1024
join=	128
gizmo=	16
líneas en el archivo stw	799

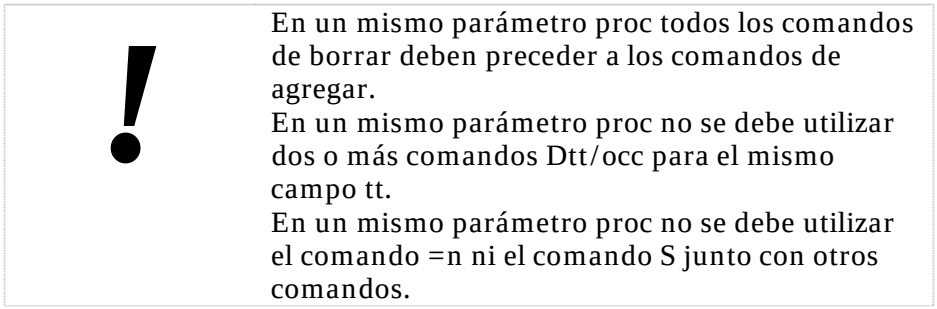
Es posible especificar hasta 1024 proc= parameters, incluyendo fst, read, write, I/F update *proc* en una línea del comando de MX. Cada *proc* sucesivo actuará sobre el registro en su situación actual, por lo que si un *proc* (o cualquier procedimiento anterior en la ejecución) modifica el registro original de entrada, el próximo *proc* actuará sobre los datos existentes en ese momento.

Se aceptan todas las instrucciones estándar de formato del CDS/ISIS, incluyendo las condicionales del tipo *IF*, pero no son aceptadas las llamadas a programas en IsisPascal (*format exits*). Acepta además las variables e0...e9, s0 ..s9 y el comando while. Es posible incluir proc dentro de proc (recursivo) sin límites.

Además, el lenguaje de formato CDS/ISIS acepta la instrucción proc( ). Para detalles, vea el Manual del lenguaje de formato CDS/ISIS.

En la tabla siguiente se describen los comandos de uso más frecuente que puede ejecutar el *proc*, los otros se explican en detalle en forma individual (MX acepta todos los comandos de la función *fldupdat( )* de la Interfaz CISIS):

Comandos de la función fldupdat ( )		
Comando	**Explicación**	**Ejemplo**
D.	Borra lógicamente el registro.	proc='d.'
D*	Borra todos los campos del registro.	proc='d*'
Dtt	Borra todas las ocurrencias del campo tt.	proc='d26'
Dtt/occ	Borra la ocurrencia occ del campo tt.	proc='d26/3'
Att#str#	Agrega la cadena de caracteres str como una nueva ocurrencia del campo tt.	proc='A999#cds#'
Htt n str_n	Agrega la cadena de caracteres str_n de n bytes de longitud como una nueva ocurrencia del campo tt.	proc='H99 8 CDS/ISIS'
=<mfn>	Cambia el número de registro (mfn) por n .	proc='=10'
S<tag>	Ordena las entradas al directorio del registro por tag.	proc='s' proc='s70'



Los comandos listados en la tabla anterior, pueden escribirse indistintamente en mayúsculas o minúsculas.

Básicamente la idea es realizar un formato que, como resultado, genere una cadena de caracteres del tipo:

```
d36a999#1999#a70#Magalhaes, A.C.#
```

Mediante la cual se borra el campo 36, se agrega 1999 al campo 999 y agrega Magalhaes, A.C. al campo 70.

Este *string* (cadena de caracteres) puede provenir de un formato como el siguiente:

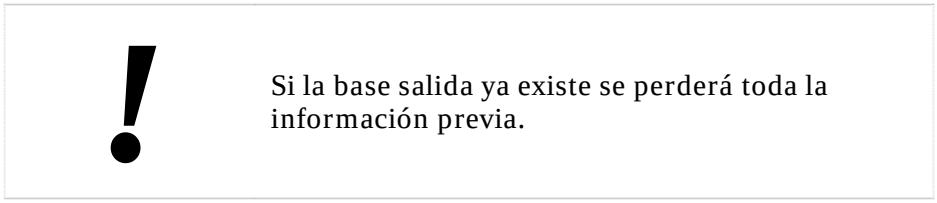
```
if p(v36) and a(v999) then 'd36a999#'v47'#'fi,"a70#"v36"#"
```

El *string* se forma a partir de la existencia del campo 36 cuyo contenido es Magalhaes, A.C., y la ausencia del campo 999.

Ejemplos:

Crear una base con todos los registros de la base origen (CDS) eliminando de éstos los campos 69 y 70:

```
mx cds proc='d69d70' create=salida now -all
```



Agregar información en el campo 999 a un grupo de registros:

```
mx cds proc='A999#1998#' from=100 to=120 copy=cds now -all
```

En este ejemplo los cambios se realizan sobre la misma base que se utilizó como fuente de entrada, por lo tanto los cambios se verán reflejados en la misma base.

Importar un archivo ISO-2709 con el MFN almacenado en el campo 999 de los registros ISO:

```
mx iso=datos.iso proc='='v999 create=master -all now
```

Exportar registros a un archivo ISO-2709 agregando el MFN del registro de origen al campo 999 de los registros ISO:

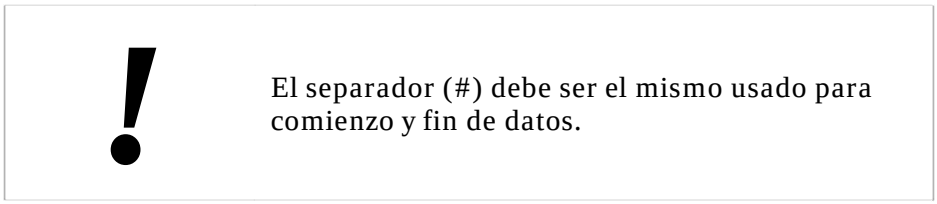
```
mx master proc='D999A999/'mfn'/' -all now iso=dato.iso
```

Función ‘A' (alta de campo) att#str#

La función ‘A' está compuesta por:

Comando	Descripción

a	Indica que se agregará un campo.
tt	Tag del campo donde se agregarán los datos.
#	Separador entre el *tt* y los datos. El separador puede ser cualquier carácter siempre que no sea numérico y que no ocurra en la cadena de caracteres a agregar.
str	Cadena de caracteres a agregar. La cadena puede ingresarse como una cadena de caracteres estática o ser extraída de un campo de datos del mismo registro o de otro (incluso de un registro de otra base de datos).
#	Separador que indica el fin de la *sting* de datos.



El separador (#) debe ser el mismo usado para comienzo y fin de datos.

El ejemplo:

```
mx cds proc='A999#1998#' from=100 to=120 copy=cds now -all
```

Es equivalente a:

```
mx cds "proc='a999{1998{'" from=100 to=120 copy=cds now -all
```

Ejemplos:

- Eliminar el campo 70, agregar sólo la primera ocurrencia de ese campo como nueva ocurrencia del campo 100.

Los siguientes ejemplos presentan dos formas de realizar la misma tarea:

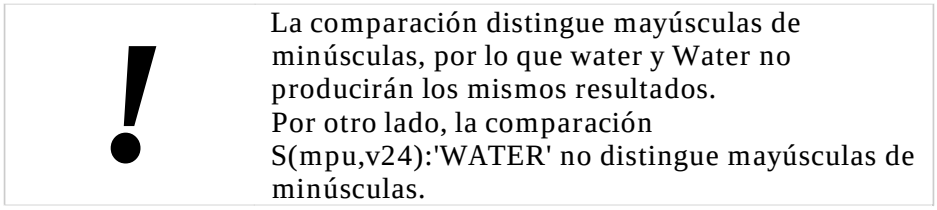
```
mx cds "proc='d70a100#',v70[1],'#' " from=10 to=20 now -all
mx cds "proc='d70',|a100@|v70[1]|@| " from=10 to=20 now -all
```

- Borrar todos los campos del registro cuando el campo 24 contiene la palabra water, y exportar el registro a una base llamada salida. El formato que contiene las instrucciones para el *proc* se obtiene de un archivo externo (prueba).

```
mx cds proc=@prueba now –all create=salida
```

El archivo prueba contiene la línea de formato:

```
If v24:'water' then 'D*' fi
```



También se puede operar sobre la misma base de datos que se utiliza como fuente de entrada.

```
mx CDS proc='d70' copy=CDS now to=1
```

- Ordenar los campos del registro por número tag:

```
mx CDS proc='s' copy=CDS now –all
```

Registro de entrada:

```
mfn= 1
44 «Methodology of plant eco-physiology: proceedings of the Montpellier Symposium»
50 «Incl. bibl.»
69 «Paper on: <plant physiology><plant transpiration> <measurement and instruments>»
24 «Techniques for the measurement of transpiration of individual plants»
26 «^aParis^bUnesco^c-1965»
30 «^ap. 211-224^billus.»
70 «Magalhaes, A.C.»
70 «Franco, C.M.»
```

Registro de salida:

```
mfn= 1
24 «Techniques for the measurement of transpiration of individual plants»
26 «^aParis^bUnesco^c-1965»
30 «^ap. 211-224^billus.»
44 «Methodology of plant eco-physiology: proceedings of the Montpellier Symposium»
50 «Incl. bibl.»
69 «Paper on: <plant physiology><plant transpiration><measurement and instruments>»
70 «Magalhaes, A.C.»
70 «Franco, C.M.»
```

Obsérvese que la salida del proceso en el ejemplo se realiza sobre la misma base de entrada de datos.

Función R<mf>,<mfn>

Agrega al archivo maestro de salida los campos del registro activo

<mf>	nombre del master de entrada de datos
<mfn>	número de registro

```
mx null proc='a10/1/a20/1/' create=xxx
10 «1»
20 «1»

mx null proc='a20/2/' append=xxx
20 «2»

mx out  proc='Rxxx,1' proc='Rxxx,2' copy=out count=1 now
mfn= 1
10 «1»
20 «1»
20 «2»
```

Graba en el master out en el registro activo (mfn=1) los campos de los registros mfn=1 y mfn=2 del master xxx.

Función <TAG[<stripmarklen>[<minlen>]]><data></TAG>

Agrega <*data*> como una nueva ocurrencia del campo <*tag*>. Las marcaciones en el estilo "<*marks*>" son eliminadas del texto hasta un máximo de <*stripmarklen*> caracteres. Si el texto que queda luego de retirar los delimitadores "<*mark*>" fuera más corto que <*minlen*> caracteres, no se efectúa la edición.

<tag>	Etiqueta del campo
<stripmarklen>	Largo máximo de la marcación (infinito por defecto)
<minlen>	Largo mínimo del campo resultante (cero por defecto)

- Agrega una ocurrencia con el término *text* en el campo 10

```
mx null "proc='<10>text</10>'"
mfn= 1
10 «text»
```

- Incluye una ocurrencia con el término *x<mark>y</mark>z* en el campo 10 eliminando las marcaciones *<mark></mark>* pero no el contenido

```
mx null "proc='<10>x<mark>y</mark>z</10>'"
mfn= 1
10 «xyz»
```

- Incluye una ocurrencia con el término *x<mark>y</mark>z* en el campo 10 eliminando la marcación *<mark>* en función del criterio de tamaño de la marcación, en este caso seis caracteres.

```
mx null "proc='<10 6>x<mark>y</mark>z</10>'"
mfn= 1
10 «xy</mark>z»
```

- Incluye una ocurrencia con el término *x<mark>y</mark>z* en el campo 10 eliminando las marcaciones *<mark></mark>* pero no su contenido pues tiene el mínimo esperado de tres caracteres.

```
mx null "proc='<10 99 3>x<mark>y</mark>z</10>'"
mfn= 1
10 «xyz»
```

- No incluye la ocurrencia con el término *x<mark>y</mark>z* en el campo 10, pues después de eliminadas las marcaciones *<mark></mark>* el contenido no llegó al mínimo esperado de cuatro caracteres.

```
mx null "proc='<10 99 4>x<mark>y</mark>z</10>'"
mfn= 1
```

Función X[{create|copy|append|merge}=]<mf>

Graba los datos del registro corriente en el archivo <out_dbn>, si no existe lo crea.

Es posible en una línea de comando asignar alternativamente diferentes archivos de salida para un mismo registro.

- Asigna como salida al master *dbn1*, luego graba el mismo registro en el mfn=3 del master *dbn2*.

```
mx null proc='a10/1/' proc='Xdbn1' proc='a20/2/' proc='=3' proc='Xdbn2'
mx dbn1
mfn= 1
10 «1»
mx dbn2
mfn= 3
10 «1»
20 «2»
```

- Crea primero un archivo *xfile.pft*, luego asigna como nombres de salida de masters a la primera ocurrencia del campo 70 de los registros 1,2,3. Observe el

formato *xfile.pft* que agrega los signos -+ al nombre. El registro mfn=1 de CDS se graba en un master "*Magalhaes*" y los registros mfn=2, mfn=3 se graban en un master "*Bosian-+*".

```
echo replace(replace(s(v70[1].8),',','-'),' ','+') >xfile.pft
mx cds proc='Xappend='@xfile.pft,  pft=mfn,x1,v70[1]/  count=3 now
000001 Magalhaes, A.C.
000002 Bosian, G.
000003 Bosian, G.
mx Magalhae    pft=mfn,x1,v70[1]/
000001 Magalhaes, A.C.
mx Bosian-+   pft=mfn,x1,v70[1]/
000002 Bosian, G.
000003 Bosian, G.
```

Función G<gizmo_mf>[,<taglist>]

Aplica un gizmo al registro, que puede ser sobre una lista de campos específicos.

```
echo transpiration~xxx> xxx.lst
mx seq=xxx.lst~  create=xxx now
mx cds proc='Gxxx,2' from=2  count=1 now
mfn=     2
24  «<The> Controlled climate in the plant chamber and its influence upon
assimilation and xxx»
30  «^ap. 225-232^billus.»
26  «^aParis^c1965»
70  «Bosian, G.»
```

Función Gsplit[/clean]=<tag>[={<char>|words|letters|numbers|trigram}]

- Ejemplo con <char> se usa el ;

```
echo Perez, J.; Garcia, María; Machado, A. > xxx.lst
mx seq=xxx.lst proc='Gsplit/clean=1=;'
mfn=     1
  1  «Perez, J.»
  1  «Garcia, María»
  1  «Machado, A.»
..
```

- Se separa el campo 44 de CDS por palabra

```
mx cds proc='Gsplit=44=words'
```

```
mfn=     1
44  «Methodology»
44  «of»
44  «plant»
44  «eco»
44  «physiology»
44  «proceedings»
44  «of»
44  «the»
44  «Montpellier»
44  «Symposium»
..x
```

Función Gsplit=<tag>=6words[/if=<if>]

Extrae las palabras de un texto en la siguiente secuencia:

- Palabra 1, palabras 1+2, palabras 1+2+3 ... palabras 1+2+...6,
- cuando finaliza la serie, se traslada a la segunda palabra
- comienza la misma secuencia.

Ejemplo:

Methodology of plant eco-physiology: proceedings of the Montpellier Symposium

```
mx cds proc='Gsplit=44=6words'
Methodology
Methodology of
Methodology of plant
Methodology of plant eco
Methodology of plant echo physiology
Methodology of plant echo physiology proccedings
of
of plant
of plant echo
of plant echo physiology
…
```

Opción /if=<if>

Hace un lookup a un archivo invertido y solamente extrae las expresiones válidas

Función Gload[/<tag>][/nonl][=<file>]

Carga el archivo <file> en formato binario en el campo <tag>

Si se indica /nonl elimina los saltos de fin de línea <CR>

Función Gdump[/<tag>][/nonl][/xml][=<file>]

Graba el contenido del campo <tag> a un archivo.

Opción /nonl: elimina los saltos de fin de línea <CR>.

Opción /xml: los graba en estructura xml.

Cambio global de patrones

gizmo=<gizmo_mf>[,<taglist>] [gizp[/h]=<out_mfx>]
[decod=<mf>]

El parámetro *gizmo* permite realizar cambios globales en el contenido de los campos de una base CDS/ISIS, convertir una cadena de caracteres en otra, y así realizar modificaciones, codificación/decodificación, compresión de datos, etc.

Estos cambios pueden realizarse sobre todos los registros de la base o sobre un conjunto de registros (seleccionados por medio de una búsqueda, un rango, etc). A su vez, los cambios pueden abarcar a todo el registro o sólo a algunos campos.

Para realizar cambios como, por ejemplo, los signos < > por / /, o la cadena de caracteres plants por PLANTS, etc., es necesario disponer de un archivo maestro *gizmo*. Este archivo maestro tiene en principio dos campos: el campo 1 contiene el dato a cambiar, y el campo 2 el nuevo valor. Cada pareja de datos será un registro del archivo maestro *gizmo*.

Cada registro de entrada se somete al procedimiento de cambio establecido en el archivo *gizmo*. Al comenzar la ejecución de MX los datos del archivo *gizmo* se cargan como una tabla en memoria y se ordenan alfabéticamente por el valor del campo 1 y por su largo (de esta manera las cadenas de caracteres más largas son convertidas antes que las cortas).

Es posible especificar hasta 16 gizmo en una línea del comando de MX. Cada *gizmo* sucesivo actuará sobre el registro en su situación actual, por lo que si un *gizmo* modifica el registro original de entrada, el próximo *gizmo* actuará sobre los datos existentes en ese momento.

Ejemplos:

- Para el ejemplo siguiente se crea una base de datos llamada *prueba* usando los parámetros de MX conocidos y se ingresan los datos directamente desde el teclado:

MS-DOS	**UNIX**

`mx seq=con create=prueba -all now`
`<|/`
`>|/`
`plants|PLANTAS`
`<ctrl>+<Z> (o <F6>)`
	`unixuser:~$ mx seq=/dev/tty1 create=prueba -all now`
	`<|/`
	`>|/`
	`plants|PLANTAS`
	`<ctrl>+<D>`

Obteniendo los registros siguientes:

```
mfn= 1
1 «<»
2 «/»
mfn= 2
1 «>»
2 «/»
mfn= 3
1 «plants»
2 «PLANTAS»
```

El contenido de los campos título y descriptores del registro MFN=1 es:

```
mx cds to=1 "pft=mfn/v24/v69"
000001
Techniques for the measurement of transpiration of individual plants
Paper on: <plant physiology><plant transpiration><measurement and instruments>
```

Al aplicar:

```
mx cds gizmo=prueba to=1 "pft=mfn/v24/v69"
```

Da como resultado:

```
000001
```

```
Techniques for the measurement of transpiration of individual PLANTAS
Paper on: /plant physiology//plant transpiration//measurement and
instruments/
```

La modificación no afecta a la base cds que provee los datos de entrada, porque ésta se produce en la salida (en este caso, la pantalla).

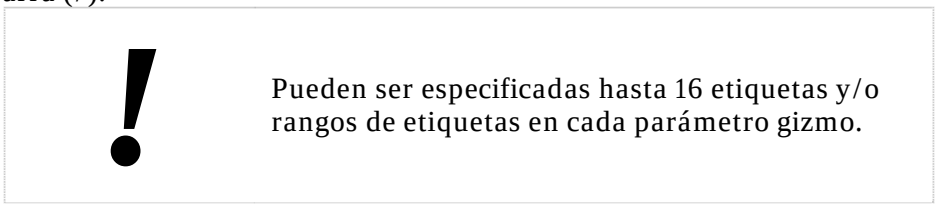
Para modificar realmente los registros se debería enviar el resultado del proceso al mismo archivo maestro, como en el ejemplo siguiente:

```
mx cds gizmo=prueba to=1 copy=cds –all now
```

Si, en cambio, se desea generar una nueva base de datos es necesario crearla:

```
mx cds gizmo=prueba to=1 create=salida –all now
```

La modificación es posible restringirla a un campo específico del registro, indicando a continuación del parámetro *gizmo* la etiqueta o etiquetas sobre las que se producirá el cambio. Es posible indicar un rango de etiquetas separadas por una barra (/).



Pueden ser especificadas hasta 16 etiquetas y/o rangos de etiquetas en cada parámetro gizmo.

```
mx cds gizmo=prueba,69,24 to=1 create=salida -all now
mx cds gizmo=prueba,35/56 to=1 create=salida -all now
```

- Otra aplicación del *gizmo* puede ser la codificación y eventual compresión de datos donde términos muy frecuentes pueden ser sustituídos por códigos o valores especiales:

```
mx seq=con create=tabla1 now -all
^aParis^bUnesco|*1*
transpiration|*2*
<ctrl>Z
```

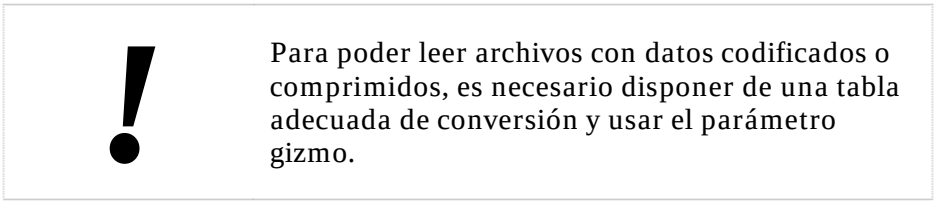
Antes de la conversión:

```
mx cds to=1 pft=mfn/v69/v24/v26
000001
Paper on: <plant physiology><plant transpiration><measurement and
instruments>
Techniques for the measurement of transpiration of individual plants
^aParis^bUnesco^c1965
```

Después de la conversión:

```
mx cds gizmo=tabla1 to=1 pft=mfn/v69/v24/v26
000001
Paper on: <plant physiology><plant *2*><measurement and instruments>
```

```
Techniques for the measurement of *2* of individual plants
*1*^c1965
```



En el ejemplo siguiente se crea la **tabla2** para realizar la conversión inversa a la **tabla1**, usando el parámetro *proc* :

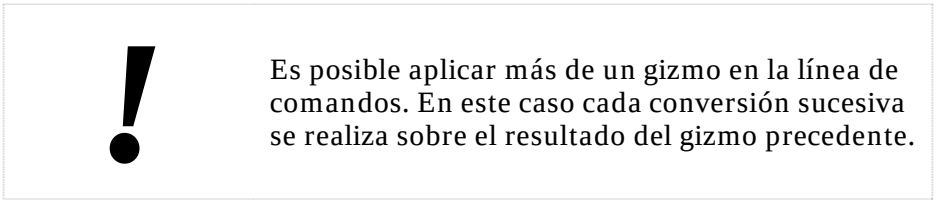
```
mx tabla1 create=tabla2 now -all "proc='d*a1#',v2,'#a2#',v1,'#'"
```

Descripción

La instrucción *proc* comienza borrando todos los campos, luego agrega como campo 1 el contenido del campo 2 y por último agrega como campo 2 el contenido del campo 1.

Teniendo las bases *tabla1* y *tabla2* se puede realizar el siguiente ejemplo:

```
mx CDS to=1 create=OUT gizmo=tabla1 now –all
mx OUT gizmo=tabla2
```



- Se aplican a la base CDS dos *gizmos* con *tabla1* y *tabla2*, que producen dos conversiones recíprocas (por lo tanto los datos originales no varían).

  ```
  mx CDS gizmo=tabla1 gizmo=tabla2
  ```

El parámetro *gizmo* tiene muchas aplicaciones. Por ejemplo, podría disponerse de tablas en varios idiomas con las equivalencias entre el código numérico de un descriptor y su versión en ese idioma. La base de datos tendrá como dato el valor numérico y la conversión a términos del idioma deseado se realizará durante el proceso de formateo.

Otro ejemplo interesante es tener una tabla con los caracteres acentuados y símbolos como campo 1 y sus códigos correspondientes en html (como ser &eacute

para é) en el campo 2, de esta manera se pueden obtener documentos html como salida.

Otra aplicación podría ser la decodificación de abreviaturas o siglas de instituciones, o abreviaturas de idiomas, de países, etc., que en el CDS/ISIS estándar se suele realizar mediante la función de ref+lookup.

Tablas de conversión mediante códigos ASCII o hexadecimales

Las tablas *gizmo* de conversión son archivos maestros compuestos de dos campos, cuyos contenidos son la cadena a cambiar (campo 1) y la cadena por la cual se va a cambiar (campo2).

Asimismo, es posible especificar dichas cadenas de caracteres a través de la representación numérica de los caracteres que las componen, permitiendo que la tabla gizmo incluya caracteres que no pueden ser tipeados o visualizados. O que no sean adecuados a la transmisión de datos en modalidad no transparente.

Este tipo de tabla puede tener hasta cuatro campos, estos son: los campos 1 y 2 (cadena a cambiar y cadena por la cual se va a cambiar) y los campos 11 y/o 21 donde se indica qué tipo de información poseen los campos 1 y/o 2 respectivamente. El valor de estos campos puede ser: **hex** (si el contenido del campo 1 o 2 está codificado en dos digitos hexadecimales) o **asc** (si el contenido del campo 1 o 2 está codificado en tres digitos decimales). Es posible ingresar comentarios en cada registro usando los campos no reservados: 1, 2, 11, 21.

Ejemplo:

Las siguientes tablas gizmo son equivalentes:

TAG	Contenido
1	OF THE
2	Ç

TAG	Contenido
1	OF THE
2	128
21	Asc

TAG	Contenido
1	079070032084072069
2	Ç
11	asc

TAG	Contenido
1	4F4620544845
2	Ç
11	hex

1. Se tiene un archivo maestro donde se desea cambiar el caracter - (hexadecimal 2D), por el caracter ~ (hexadecimal 7E).

Primero es necesario generar un tabla para realizar el cambio, que será un archivo maestro que contenga:

TAG	Contenido
1	2D
2	7E
11	Hex
21	Hex

Para esto hay que crear un archivo de texto llamado *cambio.seq*, cuyo contenido es:

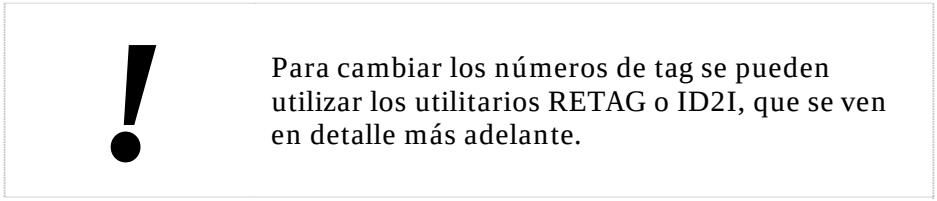
```
2D|7E|hex|hex
```

Luego se ejecuta la siguiente línea de comandos:

```
mx seq=cambio.seq create=cambio -all now
```

Con esto se generó una tabla con los cuatro campos necesarios para realizar el cambio. Aunque los campos 3 y 4 deberían tener tag 11 y 21 respectivamente, para cambiar los números de *tag* se pueden emplear los utilitarios RETAG o ID2I, pero en este ejemplo se hará con el MX usando el comando *proc*:

```
mx cambio proc='d3d4a11#hex#a21#hex#' copy=cambio
```



Para cambiar los números de tag se pueden utilizar los utilitarios RETAG o ID2I, que se ven en detalle más adelante.

Ahora está todo listo para realizar el cambio:

```
mx cds gizmo=cambio -all now
```

Estadística de la conversión por gizmo

gizp/h=<out_mfx>

El parámetro *gizp=<out_mfx>* genera un archivo maestro por cada *gizmo* utilizado en la línea de comando, con información que registra el proceso efectuado sobre los registros. Cada base tendrá como prefijo 'dbnx' más un número secuencial, por ejemplo: *dbnx0, dbnx1, dbnx2*, etc. El parámetro /h indica que los datos se guardan en hexadecimal.

Ejemplo:

```
mx cds gizmo=codigo,80 gizmo=precod,87 now –all create=OUT gizp=dbnx
```

Creará dos archivos maestros de salida: *dbnx0* correspondiente al proceso efectuado con *gizmo=codigo*, y *dbnx1* correspondiente al proceso efectuado con *gizmo=precod*.

Cada uno de esos archivos maestros tendrá un registro por cada entrada del gizmo que se haya usado efectivamente en la ejecución del MX. Los campos creados son:

TAG	Descripción
1	Campo 1 del archivo gizmo.
2	Campo 2 del archivo gizmo.
10	Cantidad de veces que fue usada esta conversión en el proceso.
31	Largo en caracteres del campo 1.
32	Largo en caracteres del campo 2.

Opción [decod=<mf>]

Nombre de un master que codifica/decodifica los datos. El archivo mf contiene el caracter separador de los elementos a decodificar y el campo donde se aplica.

Se tienen dos archivos de codificacion, *decodp* y *zdecsp*:

```
mx decodp
mfn=     1
  1  "ZDECSP"
  2  ";"
  3  "87"
```

```
  3  "88"
  3  "71"
  3  "76"
..

mx zdecsp
mfn=     2
  3  "Temefos"
mfn=     3
  3  "Matadouros"
mfn=     4
  3  "Abreviaturas"
mfn=     5
  3  "Abdome"

mx lilacs "pft=mfn/(|87=|v87/)(|88=|v88/)(|71=|v71/)(|76=|v76)#"

432938
87=^d731;^d9525^s22004;^d2465^s22004

432937
87=^d3977;^d1732^s22045;^d30912^s22045;^d7840;^d6893

432936
87=^d1942^s22004;^d2465^s22004;^d731
76=841;21044;21030
```

Unir bases de datos - JOIN

join=<mf>[:<offset>][,<taglist>]=<mfn=_fmt>

join=<db>[:<offset>][,<taglist>]=<upkey_fmt> [jmax=<n>]

El parámetro *join* permite que uno o varios registros de la misma u otras bases de datos se unan al registro que se está procesando en ese momento. El registro en proceso puede provenir de un archivo maestro ISIS, un ISO-2709 , o un archivo ASCII. En el registro de salida se agregarán todos los campos de los registros

referenciados (o campos seleccionados de éstos) y, además, se crearán campos de control con la información del proceso (numerados desde el 32001).

Al parámetro *join* se le pueden pasar los siguientes argumentos:

ARGUMENTOS DEL PARÁMETRO *join*	
Argumento	**Descripción**
`<mf> / <db>`	Una base de datos alternativa en la que se buscarán registros a unir con el registro que se está procesando. La base alternativa podría ser la misma de los datos de entrada
`[:offset]`	Desplazamiento para los tags joined
`[<taglist>]`	Una lista de los campos que serán extraídos de los registros de la base alternativa para agregar al registro de entrada. Si no se especifica la lista, entonces se agregarán todos los campos que existan en los registros hallados. Los campos de datos del registro hallado podrán renumerarse.
`<_fmt>` `@<file>`	Formato con el que se lee el registro de entrada para extraer claves con las que se buscarán registros en la base de datos alternativa. Puede ser un formato explícito o la referencia a un archivo externo.
`<upkey_fmt>`	<_fmt>|@<file> Hay que generar la(s) llave(s) de acuerdo al I/F, tipicamente en *uppercase*
`[jmax=<n>]`	Limita la cantidad de registros que se agregan al registro original procedentes de la acción del parámetro *join* para cada clave generada. Este límite se aplica para cada parámetro *join* de la línea de comandos.

Ejemplo:

```
mx cds join=autor,2,3=mhu,(v70/) create=newcds -all now
```

Descripción

1. Toma cada registro de la base de datos cds.

Por cada autor generado por la especificación de formato mhu,(v70/) realiza una búsqueda en la base *autor*.

Si encuentra algún registro en la base *autor*, agrega un campo de control y agrega los campos 2 y 3 de éste en el registro que está siendo procesado.

Por último, graba el registro en la base *newcds*.

Los siguientes diagramas esquematizan el proceso que se desarrolla cuando MX ejecuta el *join*:

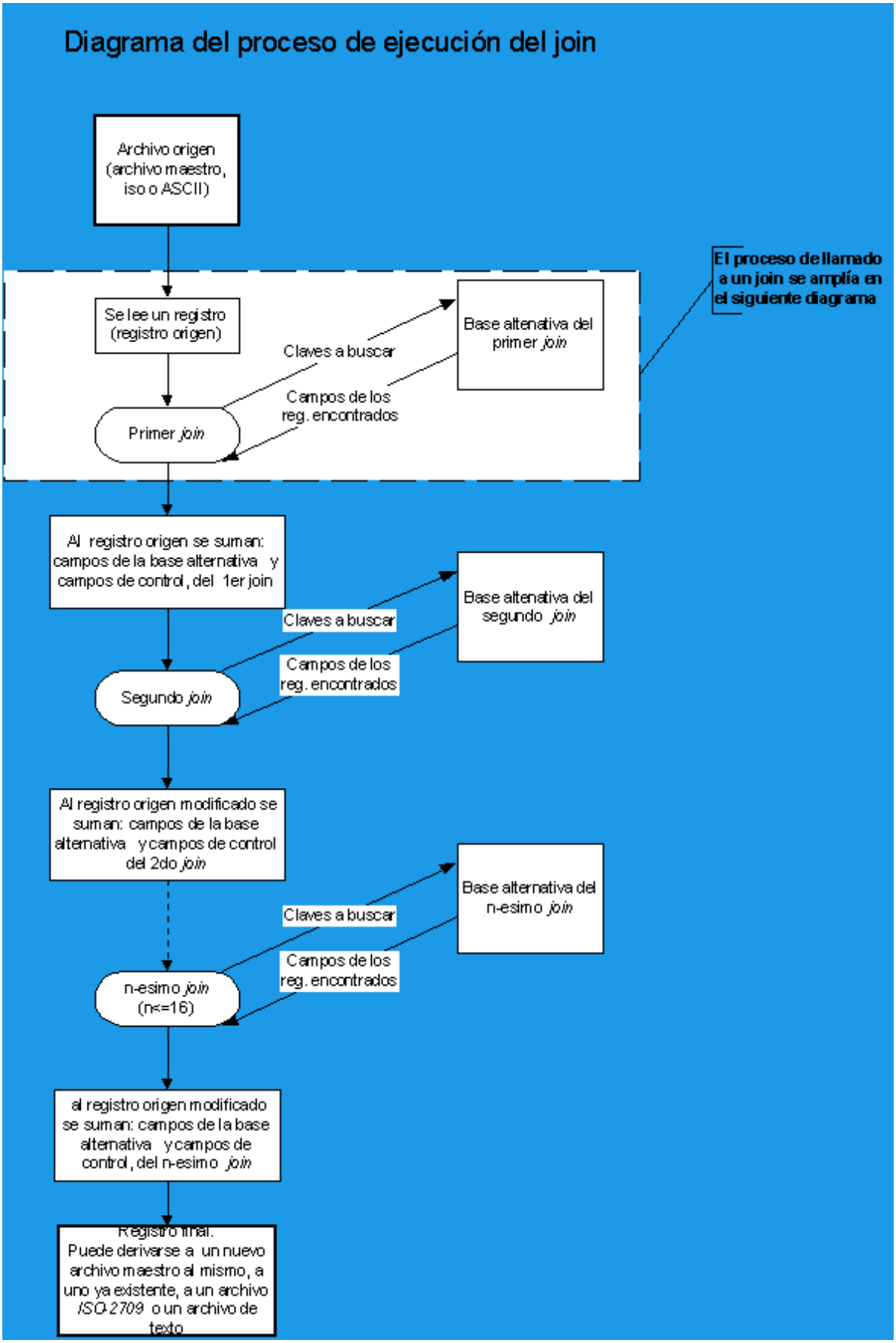
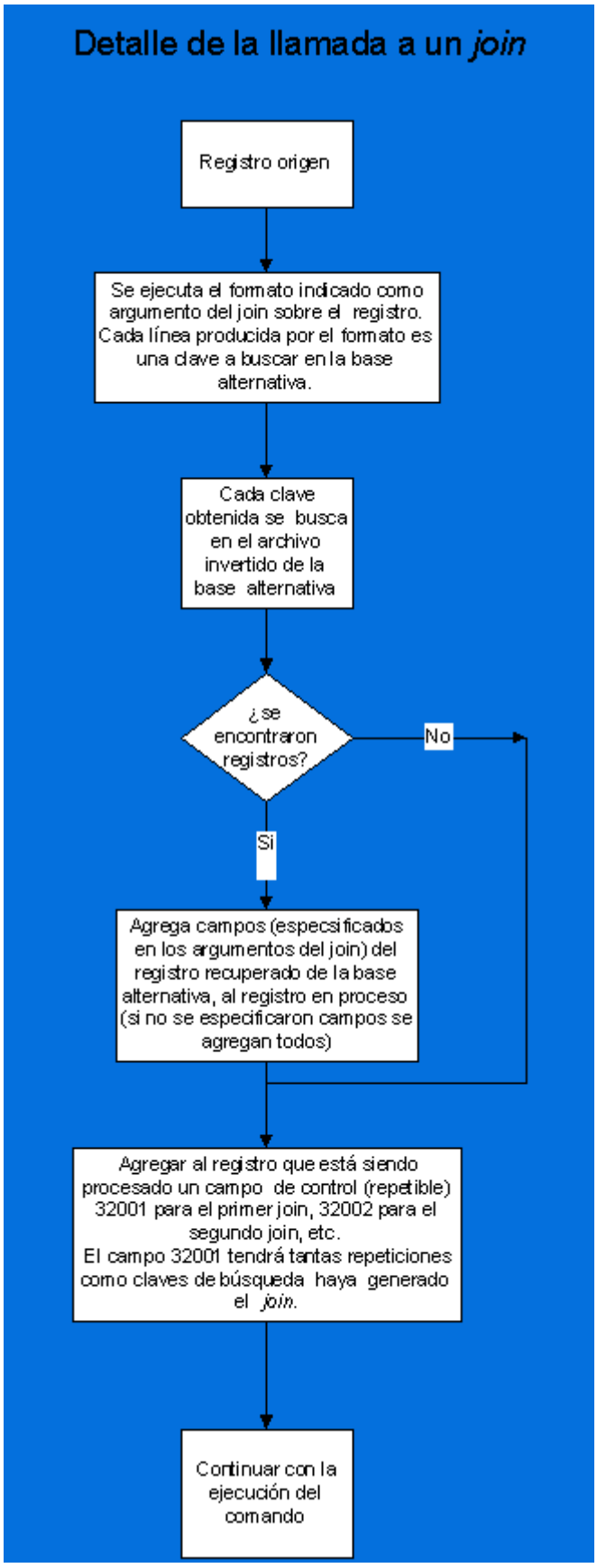
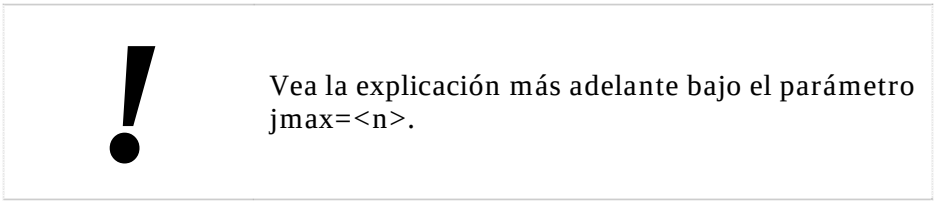


Diagrama del proceso de ejecución del join
Archivo origen (archivo maestro, iso o ASCII)
Se lee un registro (registro origen)
Claves a buscar
Base alternativa del primer join
Campos de los reg. encontrados
Primer join
El proceso de llamado a un join se amplía en el siguiente diagrama
Al registro origen se suman: campos de la base alternativa y campos de control, del 1er join
Claves a buscar
Base alternativa del segundo join
Campos de los reg. encontrados
Segundo join
Al registro origen modificado se suman: campos de la base alternativa y campos de control del 2do join
Claves a buscar
Base alternativa del n-esimo join
Campos de los reg. encontrados
n-esimo join (n<=16)
al registro origen modificado se suman: campos de la base alternativa y campos de control, del n-esimo join
Registro final. Puede derivarse a un nuevo archivo maestro al mismo, a uno ya existente, a un archivo ISO 2709 o un archivo de texto


Detalle de la llamada a un *join*
Registro origen
Se ejecuta el formato indicado como argumento del join sobre el registro. Cada línea producida por el formato es una clave a buscar en la base alternativa.
Cada clave obtenida se busca en el archivo invertido de la base alternativa
¿se encontraron registros?
No
Si
Agrega campos (especsificados en los argumentos del join) del registro recuperado de la base alternativa, al registro en proceso (si no se especificaron campos se agregan todos)
Agregar al registro que está siendo procesado un campo de control (repetible) 32001 para el primer join, 32002 para el segundo join, etc.
El campo 32001 tendrá tantas repeticiones como claves de búsqueda haya generado el *join*.
Continuar con la ejecución del comando

Es posible especificar hasta 128 *joins* en una línea del comando de MX. Cada *join* sucesivo actuará sobre el registro en su situación actual, por lo que si un *join*, o cualquier procedimiento anterior en la ejecución, modifica el registro original de entrada, el próximo *join* actuará sobre los datos existentes en ese momento.



Vea la explicación más adelante bajo el parámetro jmax=<n>.

Ejemplo:

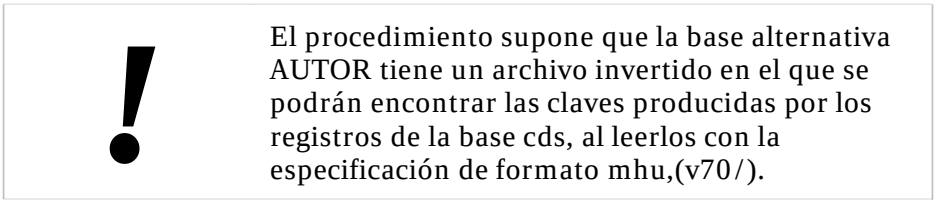
- Se dispone de una base de datos AUTOR que contiene datos normalizados de autores, como se muestra a continuación:

```
mfn= 1
1 «Magalhaes, A.C.»
2 «Ing. Agrónomo»
3 «1935-1990»
mfn= 2
1 «Franco, C.M.»
2 «Bioquímico»
3 «1940-»
...
```

Se realiza la siguiente operación sobre la base cds, cuyo campo 70 registra a los autores.

```
mx cds join=AUTOR=mhu,(v70/)
```

Entonces: se suman todos los campos de la base autor a la base cds, se agregan los campos de control 32001. Los cambios sólo se registran en la pantalla ya que no se indicó ningún destino para la salida.



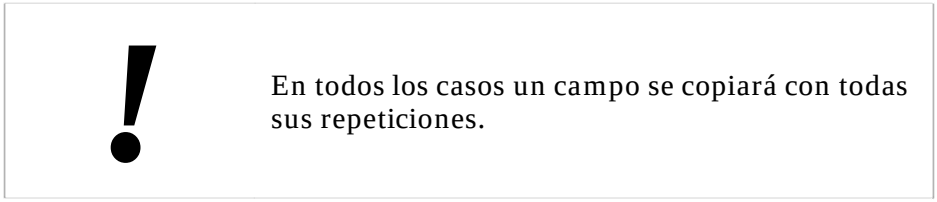
El procedimiento supone que la base alternativa AUTOR tiene un archivo invertido en el que se podrán encontrar las claves producidas por los registros de la base cds, al leerlos con la especificación de formato mhu,(v70/).

El proceso *join* envuelve los siguientes pasos:

PROCESO DEL PARÁMETRO *join*	
Descripción	**En el ejemplo**
1) Lee el registro de la base origen cds.	Base origen: cds
2) Aplica el formato especificado en el *join* al registro leido. El modo de formato MHU, o MPU, es necesario en general para obtener las claves, debido a que el archivo ISISUC.TAB estándar convierte las minúsculas en mayúsculas para la generación del archivo invertido. El registro leído es el registro proveniente de la base cds.	mhu,(v70/)
3) Para cada clave obtenida - o sea, para cada línea generada por el formato especificado - busca extraer el MFN del próximo *posting* hallado en el archivo invertido de la base de datos alternativa.	Base alternativa: AUTOR
4) Agrega al registro original un campo de control (repetible) 32001 para el primer *join*, 32002 para el segundo *join*, etc. El campo 32001 tendrá tantas repeticiones como claves distintas se hayan producido en el paso 2.	
5) Lee en la base alternativa el registro correspondiente al MFN obtenido y agrega los campos seleccionados en la lista <*tags*>. De no especificarse se tomarán todos los campos.	
6) Vuelve al paso 2.	

Lista de selección y renumeración de campos <tags>

- Selección:
  Se pueden seleccionar campos del registro alternativo de las siguientes formas:
  1. Indicando los números de campos separándolos por comas
     ```
     mx cds join=AUTOR,1,2,3=mhu,(v70/)
     ```
  2. Indicando un **rango de campos** usando valor inferior/valor superior
     ```
     mx cds join=AUTOR,1/3=mhu,(v70/) copy=cds
     ```
- Renumeración:
  Se indica el nuevo número de un campo especificando **nuevo número:viejo numero**
  ```
  mx CDS "join=AUTOR,100:1=mhu,(v70/)"
  ```



Contenido de los campos de control (32001, 32002, etc.)

Los campos con tag 32001, 32002, etc. son generados por MX para registrar el proceso del *join*.

Ejemplo:

```
32001 «AUTOR^l2^kFRANCO, C.M.^o2^m2»
```

SUBCAMPOS DEL CAMPO 32001	
Campo	**Contenido**
	Primer dato, sin indicador de subcampo: nombre de la base alternativa.
^l[1|2]	Subárbol del archivo invertido de donde se extrajo la clave: 1 para términos hasta 10 caracteres de longitud. 2 para términos entre 11 y 30.
^k	Clave con la que se hizo la búsqueda.
^o	Número de ocurrencia de la clave.
^m	MFN del registro de la base alternativa de la que proceden los datos que se agregan al registro original. Si no se encuentra el término en el diccionario de la base alternativa este subcampo no es generado.

Ejemplos:

- El siguiente ejemplo producirá la salida que se muestra más abajo. Note que el campo 1 del registro alternativo es renumerado como 100 cuando se lo copia al registro original.

```
mx cds join=AUTOR,2,3,100:1=mhu,(v70/)

mfn= 1

44 «Methodology of plant eco-physiology: proceedings of the Montpellier
Symposium»

50 «Incl. bibl.»

69 «Paper on: <plant physiology><plant transpiration> <measurement and
instruments>»

24 «Techniques for the measurement of transpiration of individual plants»

26 «^aParis^bUnesco^c-1965»

30 «^ap. 211-224^billus.»

70 «Magalhaes, A.C.»

70 «Franco, C.M.»

32001 «AUTOR^l2^kMAGALHAES, A.C.^o1^m1»
```

```
100 «Magalhaes, A.C.»
2 «Ing. Agrónomo»
3 «1935-1990»
32001 «AUTOR^l2^kFRANCO, C.M.^o2^m2»
100 «Franco, C.M.»
2 «Bioquimico»
3 «1940-»
..
```

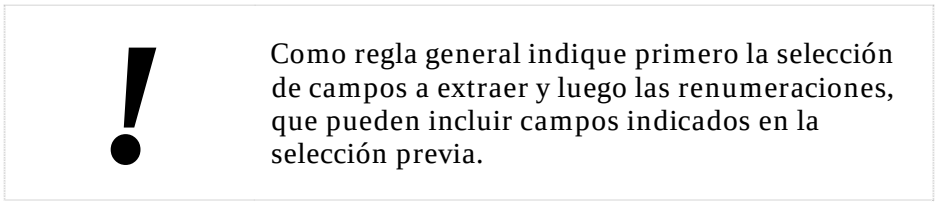
Obsérvese que se producen dos repeticiones del campo 32001, correspondientes a cada clave generada por el registro origen, y a continuación de cada una figuran los campos extraídos para cada clave de la base alternativa.

Además no se especificó el destino, por lo tanto, las modificaciones sólo se verán en pantalla.

La misma salida podría haberse producido con:

```
mx cds join=AUTOR,1/3,100:1=mhu,(v70/)
```

Donde el campo 1 se incluye en el rango 1/3, que luego es renumerado como 100.



Es importante tener en cuenta que estos campos pasan a ser campos de la base de datos. Para eliminarlos se puede utilizar el parámetro *proc*. Por ejemplo:

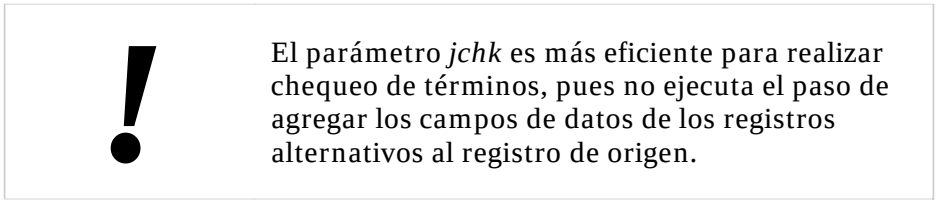
```
mx cds proc='d32001' copy=cds -all now
```

- Producir un listado de autores no válidos:

```
mx CDS join=AUTOR=mhu,(v70/) pft=@check.pft –all now
```

El archivo check.pft tiene las siguientes especificaciones de formato:

```
if p(v32001) then (if a(v32001^m) then mfn,x2,v32001^k | No existe| / fi)
fi
```



Join por número de registro

Es posible hacer un *join* por número de MFN. En este caso la especificación de formato deberá incluir, al comienzo, la cadena de caracteres *mfn* seguida por el número de registro que debe recuperarse.

El procedimiento de *join* por MFN no requiere la existencia de un archivo invertido.

Ejemplos:

- Suponiendo que el archivo llamado NUMS contiene una lista con los números de registros que se quieren extraer de la base CDS.

```
mx seq=NUMS join=CDS='mfn=',v1 create=EXPORT now -all proc='d1/1d32001'
```

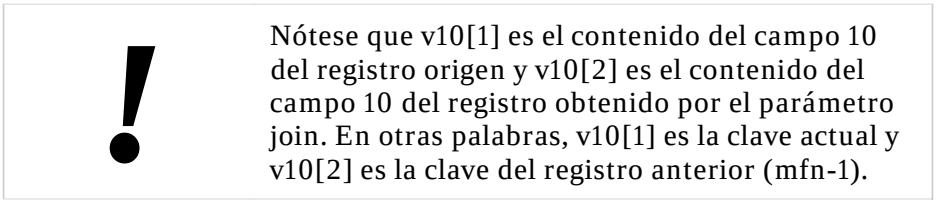
La base resultante EXPORT tiene dos campos que no están presentes en los registros de CDS. El campo 1 que proviene del archivo NUMS, donde cada línea se consideró un registro de un solo campo, y el campo 32001 que se genera para registrar el procedimiento de *join*. Estos dos campos finalmente se eliminan con el procedimiento *proc*.

Suponiendo que el archivo maestro llamado SORTED contiene sus registros ordenados segun el campo 10 (no repetible).

```
mx SORTED "join=SORTED=if mfn > 1 then 'mfn=',f(mfn-1,1,0) fi"
pft=@unavez.pft –all now
```

La siguiente especificación de formato imprime una sola vez los distintos contenidos del campo 10:

```
if v10[1] <> v10[2] then v10[1]/ fi
```



Nótese que v10[1] es el contenido del campo 10 del registro origen y v10[2] es el contenido del campo 10 del registro obtenido por el parámetro join. En otras palabras, v10[1] es la clave actual y v10[2] es la clave del registro anterior (mfn-1).

Parámetro [jmax=<n>]

El parámetro *jmax*=<*n*> limita la cantidad de registros que se agregan al registro original procedentes de la acción del parámetro *join* para cada clave generada. Este límite se aplica para cada parámetro *join* de la línea de comandos.

Si se especificara más de un *jmax*, sólo tendrá validez el último de ellos, por lo que se recomienda indicarlo una sola vez luego de detallar todos los *joins* y *jchks* necesarios para el proceso. Por defecto el valor de *jmax* es infinito.

Es posible usar hasta un máximo de 128 parámetros join y jchk en conjunto.

Confrontar Bases de datos con archivos invertidos

jchk=<if>[+<stwfile>]=<upkey_fmt>}

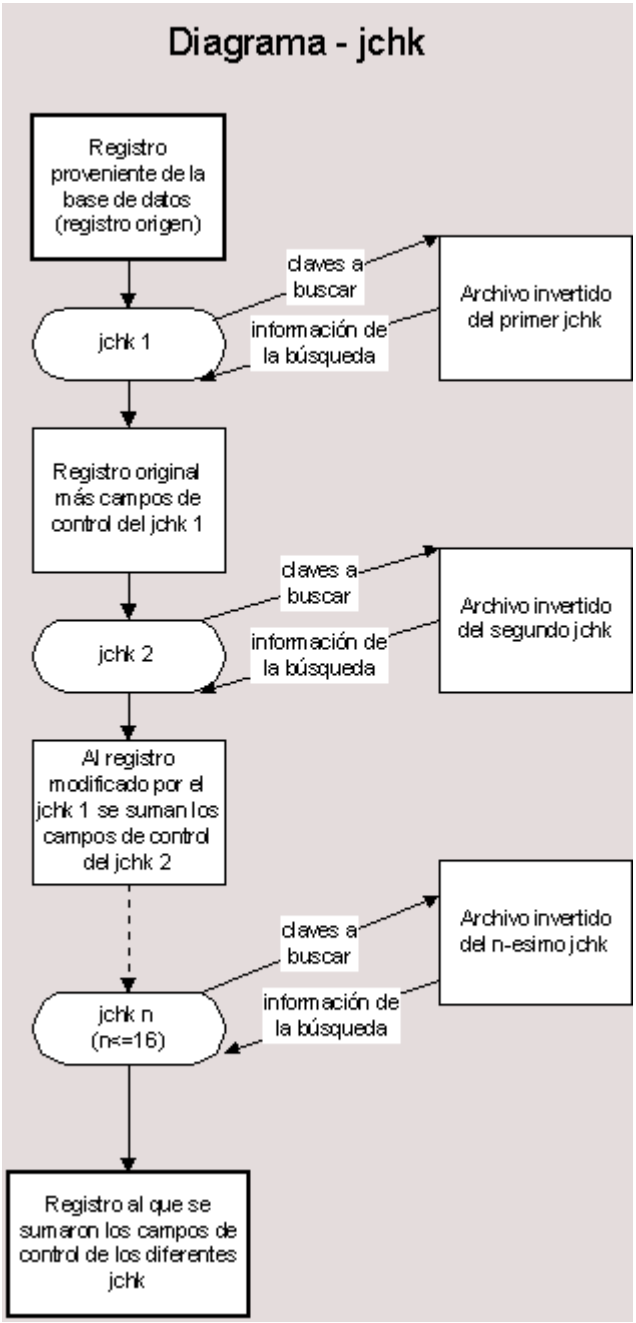
El parámetro *jchk* se usa para comparar registros (provenientes de un archivo maestro, un ISO-2709 o un archivo ASCII) contra los términos de un archivo invertido.

Al parámetro *jchk* se le pueden pasar los siguientes argumentos:

ARGUMENTOS DEL PARÁMETRO *jchk*	
Argumento	**Descripción**
`<if>`	Un archivo invertido alternativo en el que se buscarán las claves extraídas del registro en proceso.
`[+<stwfile>]`	Una lista de palabras a descartar (*stopword file*), opcional. El uso de esta opción provoca que las claves sean extraidas según la técnica de indización *palabra por palabra* (TI 4) y que cada clave generada sea filtrada por el archivo de *stopwords*.

ARGUMENTOS DEL PARÁMETRO *jchk*	
Argumento	**Descripción**
<upkey_fmt> @<file>	Formato con el que se lee el registro de entrada para extraer claves con las que se buscarán términos en el archivo invertido. Puede ser un formato explícito o la referencia a un archivo externo.

El siguiente diagrama muestra cómo se realizan los llamados a *jchk*:



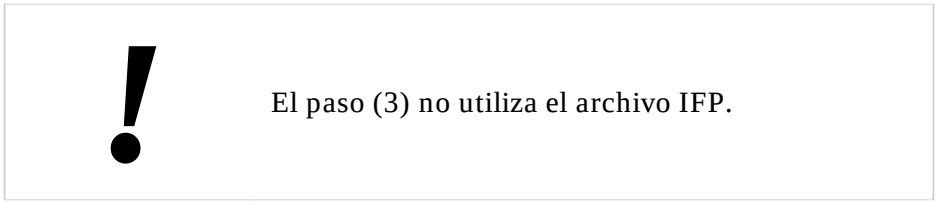
Para los ejemplos se empleará la misma base *AUTOR* que se utilizó con el *join*.

Ejemplo

```
mx CDS jchk=AUTOR=mhu,(v70/)
```

La ejecución del *jchk* está compuesta por los siguientes pasos:

EJECUCIÓN DEL *jchk*	
EXPLICACIÓN	**EN EL EJEMPLO**
Toma un registro original de la base de entrada.	Un registro de cds
Ejecuta el formato indicado como argumento del *jchk* sobre el registro origen y considera cada línea producida por ese formato como una clave de consulta. El modo de formato MHU, o MPU, es necesario en general para obtener las claves, debido a que el archivo ISISUC.TAB estándar convierte las minúsculas en mayúsculas para la generación del archivo invertido	Formato: mhu,(v70)/ Registro origen: registro que proviene de la base cds.
Para cada clave obtenida en el paso anterior busca en el archivo invertido alternativo.	Archivo invertido alternativo: AUTOR
Agrega al registro original de entrada un campo (repetible) 32001 para el primer *jchk*, 32002 para el segundo *jchk*, etc. El campo 32001 tendrá tantas repeticiones como claves distintas se hayan producido en el paso 2.	Al registro en proceso se suman los campos de control.



Si la clave está presente en el archivo invertido, *jchk* asume que es una clave válida y en el paso (4) agregará a la salida en el campo 32001 un subcampo ^m con el valor 1, en caso contrario el campo 32001 no tendrá un subcampo ^m.

Ejemplo:

El ejemplo producirá la salida que se muestra a continuación.

```
mx cds jchk=AUTOR=MHU,(v70/)
mfn= 1
44 «Methodology of plant eco-physiology: proceedings of the Montpellier
Symposium»
```

```
50 «Incl. bibl.»
69 «Paper on: <plant physiology><plant transpiration><measurement and
instruments>»
24 «Techniques for the measurement of transpiration of individual plants»
26 «^aParis^bUnesco^c-1965»
30 «^ap. 211-224^billus.»
70 «Magalhaes, A.C.»
70 «Franco, C.M.»
32001 «AUTOR^l2^kMAGALHAES, A.C.^o1^m1»
32001 «AUTOR^l2^kFRANCO, C.M.^o2^m1»
..
```

Contenido de los campos de control (32001, 32002, etc.)

El campo 32001 y los sucesivos se producen para registrar el resultado del proceso de *jchk*, y está compuesto de los siguientes subcampos:

Ejemplo:

```
32001 «AUTOR^l2^kFRANCO, C.M.^o2^m1»
```

SUBCAMPOS DEL CAMPO 32001	
Campo	**Contenido**
	Primer dato, sin indicador de subcampo: nombre del archivo invertido alternativo;
^l[1/2]	Subárbol del archivo invertido de donde se extrajo la clave, 1 para términos hasta 10 caracteres de longitud y 2 para términos entre 11 y 30;
^k	Clave con la que se hizo la búsqueda;
^o	Número de repetición del campo del registro original del cual procede la clave.
^m1	Si la clave existe se crea un subcampo *m* que contiene: "1". Si la clave no existe en el archivo invertido alternativo, el subcampo ^m no se crea.

Ejemplo:

1. Producir un listado de autores no válidos:

```
mx CDS jchk=AUTOR=mhu,(v70/) pft=@check.pft –all now
```

El archivo *check.pft* tiene las siguientes especificaciones de formato:

```
if p(v32001) then (if a(v32001^m) then mfn,x2,v32001^k | No existe| / fi)
fi
```

Nótese que este ejemplo es igual al realizado con *join* pero, como no es necesario realizar un *ref* para cada término y recuperar el registro, el tiempo de ejecución es menor.

Ventajas del *jchk* respecto del *join*

La ejecución es mucho más rápida, debido a que no es necesario acceder a cada registro; es suficiente saber que el término existe.

Otra ventaja es que *jchk* no usa el archivo .IFP. En el caso de distribuir archivos de validación (*authority files*) no es necesario entregar ese archivo, lo que implica un ahorro de espacio en disco.

> **!** Es posible especificar hasta *16 jchk* y *join* en una línea del comando de MX. Véase además: jmax=<n> en la sección *Parámetros que modifican registros* del capítulo *Parámetros que realizan procesos sobre la entrada*.

Tablas para conversión de caracteres

convert=ansi

El parámetro *convert=ansi* convierte los caracteres ASCII decimales en los caracteres alfanuméricos definidos en una tabla interna para dicho procedimiento. Por defecto CISIS usa caracteres ASCII definidos para la version IBM PC, según el código de página instalado en el computador.

Tablas para definición de caracteres alfanuméricos

[actab=<file> |ansi]

Mediante el parámetro *actab* se indica la tabla de códigos ASCII decimal para los caracteres alfanuméricos que serán considerados como parte de una palabra. Esta tabla se usa para la técnica de indización (TI) palabra por palabra (TI 4, 8, 1004, 1008). *actab=ansi* utiliza una tabla interna para dicho procedimiento.

Si no se indica este parámetro, CISIS usa en forma predeterminada la tabla correspondiente estándar del CDS/ISIS (ISISAC.TAB) definida para la versión IBM PC.

Tablas para conversión de caracteres alfabéticos a mayúsculas

[uctab=<file>|ansi]

Mediante el parámetro *uctab* se indica la tabla de conversión de los 256 caracteres ASCII a sus correspondientes en mayúsculas.

Si no se indica este parámetro, CISIS usa en forma predeterminada la tabla correspondiente estándar del CDS/ISIS (ISISUC.TAB) definida para la versión IBM PC. *uctab=ansi* utiliza una tabla interna para dicha conversión.

Tabla de Selección de Campos - generación de claves - fst

fst[/h]={<fst>|@[<file>]} [stw=@[<file>]]

ln{1|2}=<out_file> [+fix[/m]]

El parámetro *fst* extrae las claves del registro de acuerdo a la especificación de la tabla *FST* y el archivo opcional *stopword*. Si se indican los argumentos ***ln1*** y/o ***ln2***, las claves generadas se guardan en estos archivos. Si no se indican estos argumentos las claves se agregan como nuevos campos al registro que se está procesando, con *tag* igual al identificador de la línea de *FST* de donde fueron extraídos.

En caso de existir ocurrencias con dicho *tag*, se suman como nuevas ocurrencias del campo.

Ejemplo:

```
mx cds "fst=24 4 v24"
mfn= 1
24 «Techniques for the measurement of transpiration of individual plants»
26 «^aParis^bUnesco^c-1965»
30 «^ap. 211-224^billus.»
70 «Magalhaes, A.C.»
70 «Franco, C.M.»
24 «TECHNIQUES^m1^o1^c1^l1»
24 «FOR^m1^o1^c2^l1»
24 «THE^m1^o1^c3^l1»
24 «MEASUREMENT^m1^o1^c4^l2»
24 «OF^m1^o1^c5^l1»
24 «TRANSPIRATION^m1^o1^c6^l2»
24 «OF^m1^o1^c7^l1»
24 «INDIVIDUAL^m1^o1^c8^l1»
24 «PLANTS^m1^o1^c9^l1»
```

Contenido de los subcampos:

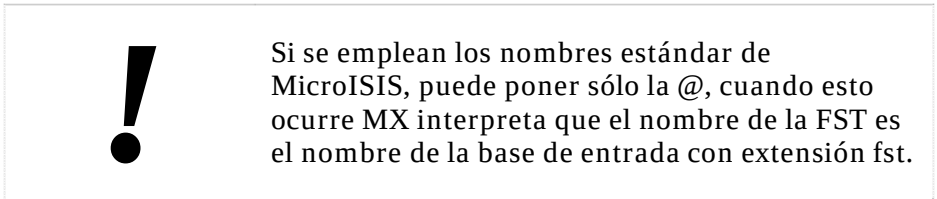
Subcampo	Contenido
CLAVE	Extraída con la FST indicada.
^m	MFN del cual se extrae la clave.
^o	Ocurrencia del campo donde aparece esa clave.
^c	Secuencia de esa clave dentro del campo.
^l[1/2]	Archivo donde se almacenará la clave: 1 para términos de hasta 10 caracteres de longitud, 2 para términos de 11 a 30 caracteres de longitud.

Referencia a una tabla de selección de campos externa

fst[/h]=@[<file>]

Se puede referenciar a un archivo externo en lugar de escribir la FST en la línea de comando:

```
mx CDS fst=@newcds.fst
```



Ejemplo:

```
mx CDS fst=@cds.fst
```

Es equivalente a:

```
mx CDS fst=@
```

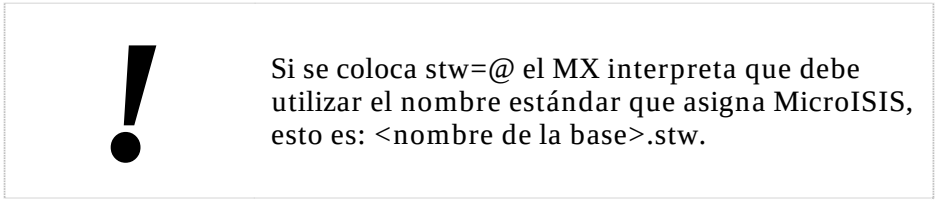
El parámetro /h permite que el formato de extracción de las claves no esté limitado a 30 (60) caracteres. Todas las claves se generan en el campo 33000, generando una ocurrencia de este campo por cada línea de extracción de la FST, no por ocurrencia de los datos originales.

Archivo de palabras no significativas (stopwords)

stw=@[<file>]

Mediante un archivo de palabras no significativas (*stopwords*) se evita la generación de términos no significativos (pronombres, preposiciones, etc.).

Un archivo de palabras no signifcativas es un archivo de texto ASCII con una palabra por línea.



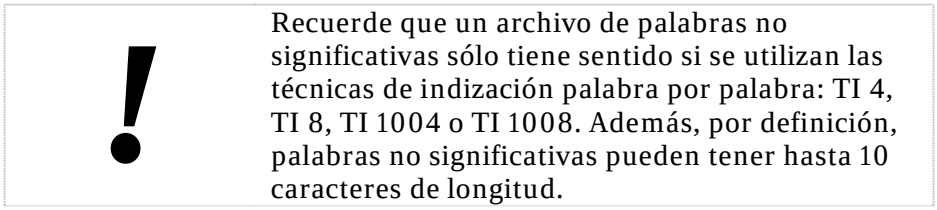
Si se coloca stw=@ el MX interpreta que debe utilizar el nombre estándar que asigna MicroISIS, esto es: <nombre de la base>.stw.

Ejemplo:

```
mx cds fst=@cds.fst stw=@
```

Es equivalente a:

```
mx cds fst=@cds.fst stw=@cds.stw
```



Recuerde que un archivo de palabras no significativas sólo tiene sentido si se utilizan las técnicas de indización palabra por palabra: TI 4, TI 8, TI 1004 o TI 1008. Además, por definición, palabras no significativas pueden tener hasta 10 caracteres de longitud.

Ejemplo:

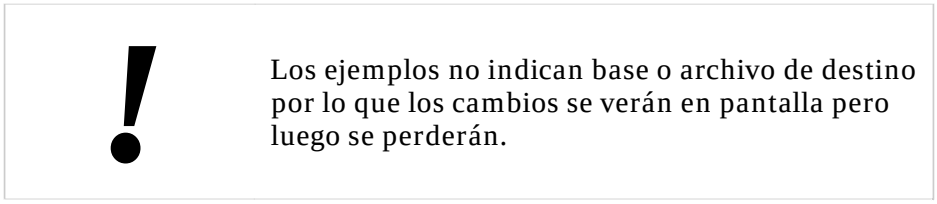
El siguiente ejemplo utiliza un archivo de *stopwords*.

El ejemplo supone la existencia de un archivo de *stopwods* (cds.stw*)* con el siguiente contenido:

```
A
AN
AND
AS
BY
FOR
FROM
IN
INTO
ITS
OF
ON
THE
TO
UPON
WITH
```

Línea de comando:

```
mx CDS "fst=24 4 v24" stw=@cds.stw
Mfn= 1
24 «Techniques for the measurement of transpiration of individual plants»
26 «^aParis^bUnesco^c-1965»
30 «^ap. 211-224^billus.»
70 «Magalhaes, A.C.»
70 «Franco, C.M.»
24 «TECHNIQUES^m1^o1^c1^l1»
24 «MEASUREMENT^m1^o1^c4^l2»
24 «TRANSPIRATION^m1^o1^c6^l2»
24 «INDIVIDUAL^m1^o1^c8^l1»
24 «PLANTS^m1^o1^c9^l1»
..
```



Los ejemplos no indican base o archivo de destino por lo que los cambios se verán en pantalla pero luego se perderán.

Técnicas de indización 1-8 / 1000 - 1008

La Interfaz CISIS acepta todas las técnicas estándar de indización (TI) de MicroISIS, (TI 0 a TI 8), a las que se suman ocho nuevas técnicas (TI 1000 - TI 1008).

Las técnicas TI 1000...1008 generan una clave compuesta de:

```
<id>   1000+IT   ‘/’, <mfn_fmt>, ’/’,  <fmt>
```

1. Cáracter delimitador de apertura.
2. El MFN.
3. Un carácter delimitador de cierre (el mismo que se utilizó para apertura).
4. La clave normal extraída según la IT 0 a 8 que corresponda.

Ejemplos:

```
1 1004  ‘|’, f(l->idif(id.10),1,0), ’|’, (v83^*)
1 1004  ‘|’, v98 ’|’, v83
```

Como resultado de esta técnica es posible controlar por programa el componente MFN del *posting* que se genera.

Ejemplo:

1. Suponiendo que, en el campo 999 del primer registro de la base cds se halla el valor 23, y que el registro 2 no tiene campo 999. Entonces la línea:

```
mx CDS to=2 "fst=1 1000 '/', if p(v999) then v999 else mfn
fi,'/',,,(|AU=|v70/) "
```

Genera la salida:

```
mfn= 1
44 «Methodology of plant eco-physiology: proceedings of the Montpellier
Symposium»
50 «Incl. bibl.»
69 «Paper on: <plant physiology><plant transpiration><measurement and
instruments>»
```

```
24 «Techniques for the measurement of transpiration of individual plants»
26 «^aParis^bUnesco^c-1965»
30 «^ap. 211-224^billus.»
70 «Magalhaes, A.C.»
70 «Franco, C.M.»
999 «23»
1 «AU=MAGALHAES, A.C.^m23^o1^c1^l2»
1 «AU=FRANCO, C.M.^m23^o1^c2^l2»
..
mfn= 2
44 «Methodology of plant eco-physiology: proceedings of the Montpellier
Symposium»
50 «Incl. bibl.»
69 «Paper on: <plant evapotranspiration>»
24 «<The> Controlled climate in the plant chamber and its influence upon
assimilation and transpiration»
26 «^c1965»
30 «^ap. 225-232^billus.»
70 «Bosian, G.»
1 «AU=BOSIAN, G.^m2^o1^c1^l2»
..
```

Obsérvese que en el registro MFN=1 los *postings* para los autores se indican como procedentes del registro MFN=23, en cambio el registro MFN=2 genera un *posting* estándar

Ejemplo

- Indizar las palabras de un texto externo asociado a un registro

```
1 1004 '/',mfn,'/' proc('Gload=archivo.txt'=v1)
```

El tamaño máximo del texto externo es de 32 kb.

Generación de archivos de ligas (links)

ln1=<out_file>/ ln2=<out_file>

Si al parámetro *fst* se agregan los argumentos *ln1* y/o *ln2* entonces se generarán los archivos de ligas (*links*) correspondientes.

Si se desea que los archivos tengan las extensiones estándar de CDS/ISIS deberán indicarse expresamente. En CISIS los archivos pueden tener o no las extensiones estándar de ISIS (pft, fst, ln1, ln2, etc.).

Ejemplos:

```
mx CDS "fst=24 4 v24" ln1=cds.ln1 -all to=10 now
mx CDS fst=@cds.fst ln2=cds.ln2 -all to=10 now
mx CDS fst=@otra ln1=cds.ln1 ln2=cds.ln2 -all to=10 now
mx CDS fst=@otra ln1=otro.1 ln2=otro.2 -all to=10 now
```

Archivos de ligas de longitud fija

Opción +fix[/m]

La opción *+fix[/m]* genera archivos .ln[1/2] de longitud fija, no estándar en CDS/ISIS, que sólo podrán procesarse con utilitarios del CISIS. Los archivos de longitud fija requieren más espacio en disco, pero pueden ser ordenados en tiempos más cortos.

La salida estándar de *CDS.LN1*, reconocida por *CDS/ISIS* es:

1 24 1 1 TECHNIQUES
1 24 1 8 INDIVIDUAL
1 24 1 9 PLANTS

La salida especial de longitud fija tiene la estructura:

CLAVE MFN TAG OCC CNT

```
mx CDS "fst=24 4 v24" ln1=cds.ln1 +fix
```

Que producirá el archivo siguiente:

TECHNIQUES	1	24	1	1
INDIVIDUAL	1	24	1	8
PLANTS	1	24	1	9

Con la opción *+fix/m* se generarán archivos .ln[1/2] de longitud fija solamente con la CLAVE y el MFN del *posting*.

Los archivos de ligas con CLAVE y MFN, sin los componentes TAG y OCC, son indicados para aplicaciones que manejan más de un archivo invertido generados para un mismo archivo maestro.

Salida

Ejecución de programa externo

sys[/show]={<sys_fmt_spec> | @<file>}

El parámetro *sys=* provoca la ejecución de un comando del sistema operativo, el que debe especificarse como una instrucción de formato. El resultado de este formato debe producir comandos válidos para el sistema operativo.

Cada línea generada por el formato será ejecutada como un comando del sistema operativo.

Ejemplo:

Suponiendo que en el campo 777 de los registros de la base de datos figura el nombre de un archivo de texto (alltext1.txt) que se quiere mostrar, y que contiene texto ASCII (el texto completo de un trabajo, por ejemplo):

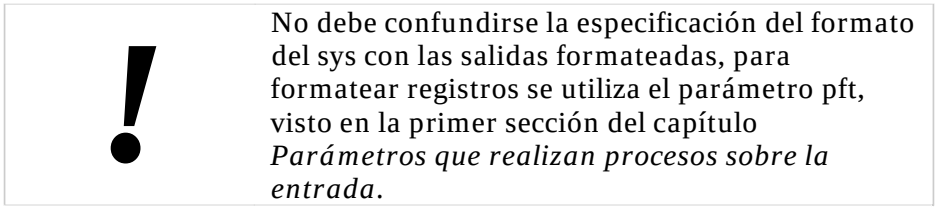
Agregar al registro 1 de la base cds el campo 777 con contenido alltext1.txt

```
mx cds to=1 proc='a777#alltext1.txt#’  copy=cds
```

Crear un archivo alltext1.txt con algún contenido

Ejecutar el MX indicando que muestre el archivo alltext1.txt con el parámetro *sys*

```
mx cds "sys=if p(v777) then 'more 'v777 fi"
```



El ejemplo siguiente es incorrecto:

```
mx cds "sys=if v24:'water' then mfn/  ‘dir *.mst'  fi"
```

Debe indicarse como:

```
mx cds "pft=if v24:'water' then mfn/ fi"  "sys=if v24:'water' then ‘dir
*.mst ' fi"
```

Opción */show*

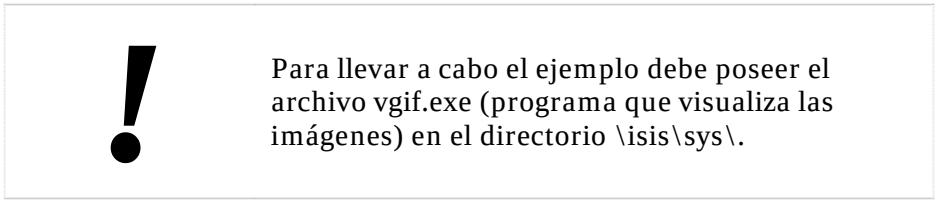
La opción */show* detiene la ejecución del comando indicado en *sys*, muestra el comando que se va a ejecutar, y espera la intervención del operador para seguir adelante.

```
mx cds "sys/show=if v24:'water' then ‘dir *.mst' fi"  to=20 now
```

Una aplicación del parámetro *sys* podría ser ejecutar un programa externo sobre los datos de un registro, para visualizar, por ejemplo, una imagen cuyo nombre de archivo está almacenado en un campo.

Ejemplo:

```
mx cds -all "sys=|\isis\sys\vgif.exe |v100^a"
```



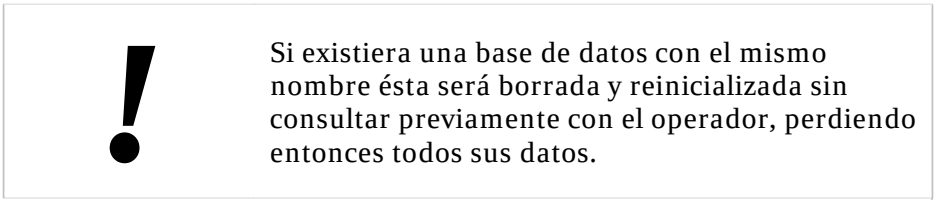
Parámetros que crean/modifican bases de datos

{create | copy | append | merge | updatf}=<out_dbn>

Creación de un archivo maestro

create=<out_dbn>

El parámetro *create* crea e inicializa incondicionalmente un archivo maestro con el nombre asignado en <out_dbn>, archivo maestro en el que se guardarán los registros resultantes del proceso.

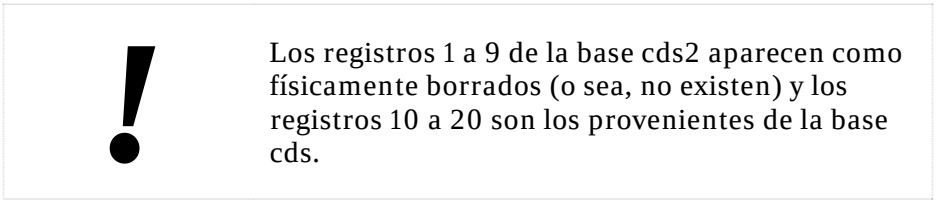


Si existiera una base de datos con el mismo nombre ésta será borrada y reinicializada sin consultar previamente con el operador, perdiendo entonces todos sus datos.

Ejemplos:

- El siguiente ejemplo lee los registros 10 a 20 de la base cds y los graba en la base cds2;

```
mx cds from=10 to=20 create=cds2 now -all
```



Los registros 1 a 9 de la base cds2 aparecen como físicamente borrados (o sea, no existen) y los registros 10 a 20 son los provenientes de la base cds.

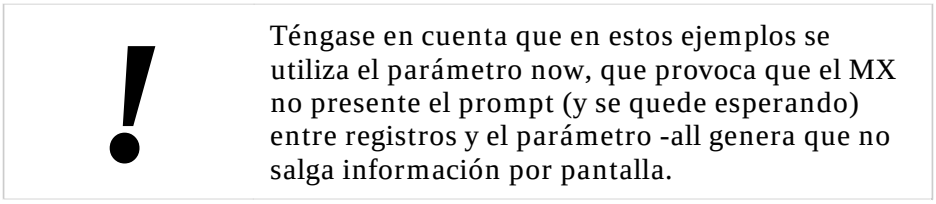
- Crear una base con los registros resultantes de una búsqueda:

```
mx cds "plants and water" create=plawat now -all
```

Los registros de salida son grabados con los mismos MFNs de la origen.

- Crear una base a partir de un archivo ISO-2709 (debería existir el archivo cds.iso):

```
mx iso=cds.iso create=cds3 now -all
```



Copiar registros a un archivo maestro

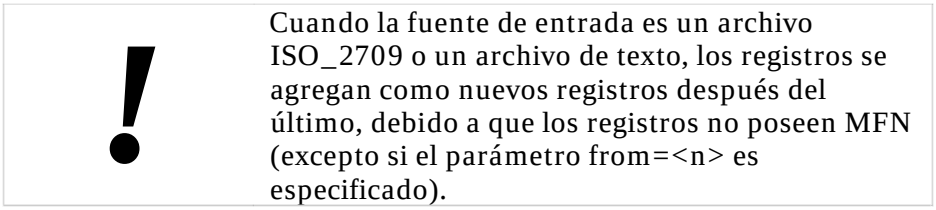
copy=<file>

El parámetro *copy* escribe los registros procesados en el archivo maestro de salida <out_dbn> con el MFN que poseía el registro que fue leído. Si en <out_dbn> ya existe un registro con ese mfn se perderá su contenido y si el registro no existe, será creado.

Cuando la base de salida (<out_dbn>) es la misma que la de entrada, *copy* funciona como si modificara los registros, ya que estos son leidos, modificados y grabados en la misma base con el mismo mfn.

A diferencia del *create*, el *copy* no reinicializa la base destino.

Si la base destino no existe, es creada. En este caso *copy* funciona exactamente igual que *create*.



Ejemplos:

- Copiar los registros 30 a 40 de la base cds en la base cds2:

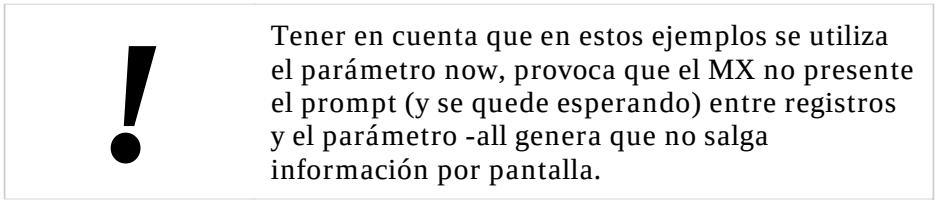
```
mx cds copy=cds2 now from=30 to=40 -all
```

- Copiar el resultado de una búsqueda en una nueva base de datos:

```
mx cds water copy=cds2 now -all
```

- Leer registros de un archivo ISO-2709 y guardarlos en una base ya existente:

```
mx iso=cds.iso copy=cds2 now -all
```



Agregar registros a una base de datos

append=<dbn_out>

El parámetro *append* guarda los registros procesados como nuevos registros en la base de salida a continuación del último existente. A diferencia de *create* y *copy*, los registros procesados pierden su número de mfn original.

Si la base destino no existe es creada.

Ejemplos:

- Agregar los registros 30 a 40 de la base cds en la base cds2:
  ```
  mx cds append=cds2 now from=30 to=40 -all
  ```
- Agregar el resultado de una búsqueda a una nueva base de datos:
  ```
  mx cds "plants * water" append=cds2 now -all
  ```
- Leer registros de un archivo ISO-2709 y guardarlos en una base ya existente:
  ```
  mx iso=cds.iso append=cds2 now -all
  ```

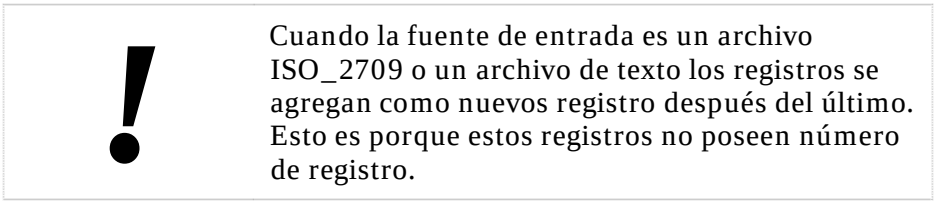
Mezclar/Intercalar registros

merge=<outdbn>

El parámetro *merge* escribe los registros procesados en la base de salida solamente en el caso en que esos registros no existan en <out_dbn>.

Suponiendo que en la base <out_dbn> existen todos los registros entre MFN=1 y MFN=20 con la excepción del registro MFN=10, y que <in_dbn> está compuesta de 15 registros con MFN 1 a 15. Si aplicamos un merge, la base resultante estará compuesta de 20 registros de los cuales el 10 proviene de <in_dbn> y el resto son los que estaban ya en <out_dbn>.

Si la base de salida <out_dbn> no existe, es creada.



Cuando la fuente de entrada es un archivo ISO_2709 o un archivo de texto los registros se agregan como nuevos registro después del último. Esto es porque estos registros no poseen número de registro.

Ejemplos:

- Insertar los registros de cds, con mfn entre 1 y 20, que no estén en cds2:

```
mx cds merge=cds2 now from=1 to=20
```

- Insertar los registros resultado de una búsqueda hecha en cds que no estén en cds2:

```
mx cds merge=cds2 energy$ -all now
```

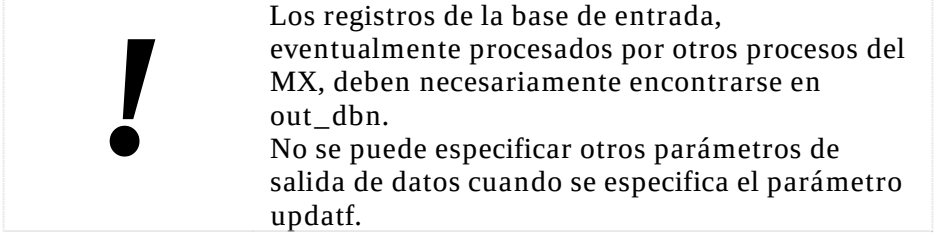
Actualización de campos

updatf=<out_dbn>

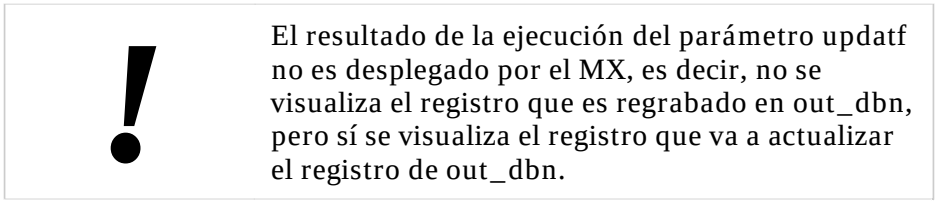
El parámetro *updatf* (*update fields*) toma los datos del registro de entrada y sustituye los campos del registro de salida, en el caso que coincidan los *tags*.

out_dbn es la base que tiene los registros con campos sin actualizar y que seran regrabados. El proceso de actualización que realiza MX sobre *out_dbn* es el siguiente:

1. Lee un registro de la base de entrada y ejecuta los procesos de *gizmo*, *join*, *proc*, *fst*, si fueron especificados;
2. Busca un registro con el mismo mfn en *out_dbn*, con el siguiente resultado:
   - todos los campos del registro leído de *out_dbn*, que existan en el registro leído de la base de entrada, son reemplazados por éstos,
   - todos los campos del registro leído de *out_dbn*, que no existan en el registro leído de la base de entrada, son mantenidos,
3. Regraba el registro en *out_dbn*.



Los registros de la base de entrada, eventualmente procesados por otros procesos del MX, deben necesariamente encontrarse en out_dbn.
No se puede especificar otros parámetros de salida de datos cuando se especifica el parámetro updatf.



Ejemplos:

Se tienen dos bases de datos:

INPUT	CDS
Mfn= 1 1 «campo 1 nuevo» 2 «campo 2 nuevo» 4 «campo 4 nuevo»	Mfn= 1 44 «Methodology of plant eco-physiology: proceedings 69 «Paper on: <plant physiology><plant transpiration> 24 «Techniques for the measurement of transpiration 26 «^aParis^bUnesco^c-1965» 30 «^ap. 211-224^billus.» 70 «Magalhaes, A.C.» 70 «Franco, C.M.» 1 «campo1» 1 «campo 1b» 2 «campo 2» 2 «campo2b» 2 «campo 2c» 3 «campo3»

El siguiente ejemplo realiza la actualización de campos de los registros de CDS tomando como fuente los registros de la base INPUT

```
mx INPUT updatf=CDS -all now
mx CDS
```

Este es el resultado de un registro de la base CDS con los cambios efectuados:

```
mfn= 1
44 «Methodology of plant eco-physiology: proceedings
69 «Paper on: <plant physiology><plant transpiration>
24 «Techniques for the measurement of transpiration
26 «^aParis^bUnesco^c-1965»
30 «^ap. 211-224^billus.»
70 «Magalhaes, A.C.»
```

```
70 «Franco, C.M.»
3 «campo3»
1 «campo 1 nuevo»
2 «campo 2 nuevo»
4 «campo 4 nuevo»
```

Obsérvese que los campos 1 y 2 de CDS fueron sustituídos por los procedentes de la base INPUT, pero los otros campos (incluyendo el campo 3) no fueron modificados y además se agregó el campo 4.

Generar un Archivo ISO_2709

[out]iso[={marc|<n>}]=<out_isofile> [outisotag1=<tag>]

El programa MX puede leer y escribir indistintamente archivos en formato ISO-2709 y aplicar los mismos procedimientos que sobre los archivos .MST/XRF (exceptuando aquellos procesos que requieren el uso del archivo invertido o diccionario).

La opción *iso=<out_isofile>* o *iso=marc|<n>=<out_isofile>* permite generar archivos ISO-2709. De la misma manera que como resultado de un proceso se genera un archivo maestro, se puede crear un archivo ISO-2709.

Con la opción *iso=<n>=<out_isofile>* puede indicarse un largo fijo para las líneas. Esta opción se utiliza para intercambiar archivos .ISO de PCs con computadoras de mayor porte como el HP que utiliza el programa MinISIS. El separador de campo y el separador de registro en el archivo generado por el MX es el carácter #. Indicando *iso=0=<filename>* las lineas de los registros serán de largo variable.

La opción *iso=marc=<out_isofile>* generará líneas de salida con las siguientes características compatibles con la definición de MARC:

- Líneas de largo variable
- Los separadores de campo y registro serán los caracteres ASCII '\029' '\030'
- Las lineas terminarán solamente con '\012' y no incluirán el '\013' (CR).

Ejemplos:

- Este ejemplo produce un archivo ISO-2709 de la base CDS

```
mx CDS iso=salida.iso -all now
```

- Toma como entrada una base de datos (CDS), realiza una búsqueda (plants and water) y genera con los registros recuperados un archivo ISO-2709 como salida (cds.iso).

```
mx cds "plants and water" iso=cds.iso now -all
```

- Este ejemplo genera un archivo ISO-2709 con el registro 1 de la base CDS, con ancho de línea fijo=40.

```
mx cds iso=40=salida.iso to=1 now -all
```

- Crea una salida del tipo:

```
00409000000000109000450004400780000000690
07900078024006900157026002300226030002l0
02490700016002700700013002861Methodology
of plant eco-physiology: proceedings of
the Montpellier Symposium1Paper on: <pl
ant physiology><plant transpiration><mea
surement and instruments>1Techniques for
the measurement of transpiration of ind
ividual plants1^aParis^bUnesco^c-1965#^a
p. 211-224^billus.1Magalhaes, A.C.1Franc
o, C.M.11
```

> **!** Obsérvese que la ultima línea es llenada con blancos a la derecha.

Recuerde que un archivo ISO-2709 se puede leer con MX, indicando solamente que será la fuente de entrada:

```
mx iso=cds.iso
```

Generar un Archivo ASCII con separadores

fix=<out_file>

Despliega cada registro en una sola línea en formato ASCII puro, separando todos los campos por la barra vertical (|). Toda la información sobre etiquetas se pierde. El ancho máximo de la línea está determinado por el sistema operativo.

Ejemplo:

- Se dispone de una base de datos llamada ADMIN que contiene los datos de editorial, título y precio de cada libro. La instrucción siguiente genera el archivo de salida PRECIOS:

```
mx admin fix=precios now -all
```

Archivo PRECIOS**:**

Editorial 1|Titulo 1|124
Editorial 1|Titulo 2|100
Editorial 2|Titulo 3|89
Editorial 2|Titulo 4|99
Editorial 2|Titulo 5|101.50

Este archivo puede importarse a otros programas que admitan campos de datos separados por delimitadores. Por ejemplo, puede importarse a una planilla electrónica del tipo Excel para realizar cálculos y/o análisis de tipo estadístico, económico, etc.

Intercambiar datos del Leader del registro

[[out]isotag1=<tag>]

Los registros MARC almacenan datos en las posiciones 5-8 y 17-19 del la cabecera del registro (leader). Las instrucciones de formato de CISIS no tienen acceso directo a esas posiciones, pero es posible convertir esos bytes en campos convencionales del archivo maestro tanto al ingreso como en la exportación de registros.

En el proceso de ingreso los datos del leader se cargan en una serie consecutiva de campos a partir de un valor base + la posición del byte del leader. Por ejemplo, si se designa como base el campo 3000, entonces el byte 5 del leader se cargará en el campo 3005. Durante la exportación, se tomarán de esos campos los valores que hubieren para registrarlos en los lugares correspondientes del leader. Si no hubiera dato en el campo se pondrá un espacio en blanco en la posición correspondiente del leader.

Ejemplo

```
mx mrclte
mfn=     1
  1  «00089048230 /AC/r91»
  3  «DLC»
  8  «891101s1990    maua   j      001  0eng  »
 10  «##^a###89048230 /AC/r91»
 20  «##^a0316107514»
 40  «##^aDLC^cDLC^dDLC»
 50  «00^aGV943.25^bB74 1990»
 82  «00^a796.334/2^220»
100  «##^aApter, David Ernest^d1919-1999»
245  «10^aMake the team.^pSoccer :^ba heads up guide to super soccer! /^cRi
J. Brenner.»
250  «##^a1st ed.»
260  «##^aBoston :^bLittle, Brown,^cc1990.»
300  «##^a127 p. :^bill. ;^c19 cm.»
3005  «c»
3006  «a»
3007  «m»
3017  «5»
3018  «i»
```

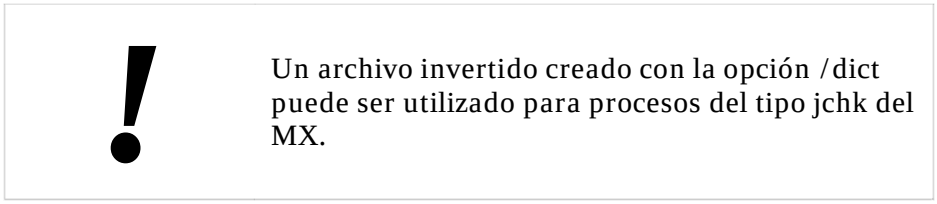
Cargar elementos generados por una FST

Funcion fullinv

Opción /dict

La opción */dict* crea sólo el diccionario del archivo <out_inf>, no crea contenido en el archivo out_inf.ifp.

```
mx CDS fst=@ ifupd/create/dict=cds now from=10 to=30
```



Un archivo invertido creado con la opción /dict puede ser utilizado para procesos del tipo jchk del MX.

Opción /m

Extrae las llaves (actualiza el I/F) solo con el componente MFN del posting

Opción /ansi

Cuando se indica **ansi**, se crea una tabla interna padrón para el conjunto de caracteres ANSI, liberando de la necesidad de emplear los parámetros **actab** y **uctab**

Tabulación de frecuencia

tab[/lines:100000/width:100/tab:<tag>]=<tab_fmt>

Genera una tabulación de frecuencia del contenido de un campo/subcampo. El valor tabulado es la diferencia con 1.000.000.000 (*high-value*), lo que permite la ordenación ascedente/descendente.

Grabar el contenido del tag <tag> en un archivo de nombre <mfn>.<tag>

Ejemplo

```
mx cds  "tab=(v26^c/)" now lw=0 | sort | more
999999949|50|1966
999999962|37|1976
999999974|25|1965
999999991|8|1973
999999994|5|1975
999999995|4|1974
999999997|2|1968
999999997|2|1971
999999997|2|27 Aug. 1976
999999998|1|11 Dec. 1975
999999998|1|14 June 1976
```

```
999999998|1|15 June 1976
999999998|1|17 Sept. 1976
999999998|1|1983
999999998|1|23 Oct. 1975
999999998|1|25 June 1976
999999998|1|6 Feb. 1976
```

Parámetros de inicialización / variables de ambiente (setup)

Estos parámetros permiten cambiar los valores que MX toma por defecto tales como: el tamaño máximo de un registro, el de una salida de formato, la definición de variables de ambiente o de nombres lógicos, etc.

Archivo de parámetros CISIS

cipar=<archivo>

El archivo CIPAR es una herramienta que provee las funciones del SYSPAR.PAR y de los <dbn>. par del CDS/ISIS estándar; además agrega otras funciones.

El CIPAR es un archivo ASCII puro que –entre otras funciones– une nombres lógicos definidos en el CIPAR a los nombres físicos (reales) de los archivos, de esta manera se independiza del nombre real y de su ubicación.

Explicación detallada de este parámetro en el Apéndice II - Archivo CIPAR.

Ejemplo:

Si el archivo *\dir\filename* contiene:

```
cds.*=\cisis\bases\cds.*
cds1.pft=\cisis\bases\cds1.pft
```

El comando:

```
mx cipar=\dir\filename cds pft=@cds1.pft
```

Es equivalente a:

```
mx \cisis\bases\cds pft=@\cisis\bases\cds1.pft
```

Tamaño máximo de un registro

mfrl=<n>

Este parámetro determina el tamaño máximo que puede llegar a tener un registro en la lectura, proceso y grabación, incluyendo los *join*, *proc* y *fst*. El valor predeterminado para MX es 32767 bytes, en CDS/ISIS para MS-DOS es de 8.192 bytes y en Winisis es 64Kb

Es necesario tener cuenta que cuando los registros son procesados por los distintos procedimientos internos del MX (como *join* o *proc*) pueden llegar a tener largos transitorios superiores a ese límite.

Ejemplos:

```
mx mfrl=32000 CDS a$ join=AUTHORS=mhu,(v70/) create=OUT
```

Tamaño máximo para el resultado de un formato

fmtl=<n>

Este parámetro determina el tamaño máximo del área interna donde se almacena el resultado de todo formato especificado (*pft, sys, join,* etc). El largo predefinido es igual al largo del registro por defecto.

Ejemplo:

```
mx mfrl=20000 fmtl=20000 TITLES join=ISSUES=v30 pft=@holdgins.pft
```

Parámetros generales

Parámetros que controlan la salida por pantalla

{+ | -}{control | leader | xref | dir | fields | all }

Estos parámetros determinan la salida que ofrecerá el procedimiento de volcado (*dump*) en la salida estándar (por defecto la pantalla), de los datos, lo que es útil para revisar los archivos físicos .MST y .XRF.

La salida predeterminada es *+fields*, que implica la salida sin formato de todos los campos del registro.

Los registros borrados se incluyen en el procedimiento de volcado y se los indica con el mensaje [DELETED].

Parámetro +

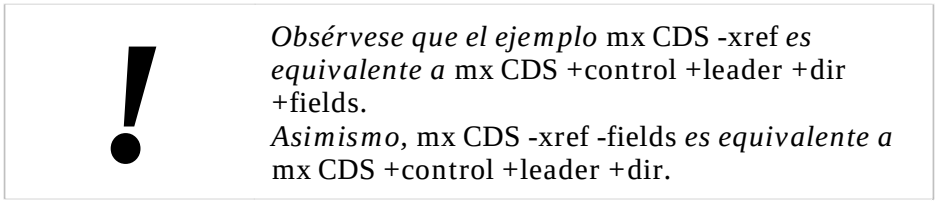
Activa la opción asociada al procedimiento de salida por pantalla. La primera vez que se lo emplea cancela las opciones predefinidas.

Parámetro -

Desactiva la opción asociada al procedimiento de salida por pantalla. La primera vez que se lo emplea activa todas las otras opciones predefinidas.

Ejemplos:

```
mx CDS +xref
mx CDS -xref
mx CDS +control +leader +dir
mx CDS +all
```



Obsérvese que el ejemplo mx CDS -xref *es equivalente a* mx CDS +control +leader +dir +fields.
Asimismo, mx CDS -xref -fields *es equivalente a* mx CDS +control +leader +dir.

Los parámetros se refieren a las siguientes partes del .MST y .XRF, a los que no se tiene acceso directo por el CDS/ISIS estándar.

Opción	Visualiza
control	Registro de control del MST, identificado como MFN=0.
leader	Segmento de longitud fija de 18 bytes al comienzo de cada MFN.
dir	Directorio dentro del registro que contiene los índices a los campos de datos.
fields	Los datos contenidos en los campos del registro.
xref	Contenido del archivo .XRF.
all	Activa o desactiva todas opciones anteriores.

Para entender mejor la utilidad de estos parámetros es imprescindible conocer la estructura de los regisros ISIS.

Un registro con estructura ISIS tiene dos características especiales que ofrecen una gran versatilidad para el manejo de información textual: campos repetibles y de longitud variables.

Puesto que los registros no tienen un largo predeterminado, ni los campos tienen un largo fijo, ni una cantidad predeterminada de repeticiones, no es posible tener acceso directo a ninguna porción de datos dentro de la base.

El acceso al registro se hace de modo indirecto, mediante punteros en un archivo auxiliar con extensión .XRF, y dentro del registro se accede a los datos mediante punteros en un directorio.

El archivo .XRF contiene toda la información necesaria para encontrar el punto de comienzo del registro solicitado dentro del .MST.

Para más detalle véase Apéndice: Estructura de los registros de una base ISIS.

Parámetros para ambientes multiusuarios

Opciones de proceso

[mono | mast | full]

Si una base de datos se actualiza en un ambiente multiusuario debe indicarse antes el parámetro *mast* o *full*. El valor *mono* (monousuario) es el valor por defecto.

Modo monousuario: mono

Es el modo por defecto, no se verifica consistencia ni acceso a los datos. Supone un único usuario.

Este parámetro se corresponde con el parámetro 14=0 del Syspar de CDS/ISIS

Acceso limitado a los datos: mast

Este parámetro permite en forma simultánea la recuperación y actualización del archivo maestro. Debe garantizarse que el archivo invertido no sea actualizado.

Este parámetro se corresponde con el parámetro 14=2 del Syspar de CDS/ISIS

Acceso completo: full

Cuando este parámetro está presente se puede recuperar y modificar datos en simultáneo, tanto en el archivo maestro como en el invertido. Por otra parte, el sistema se vuelve más lento.

Este parámetro se corresponde con el parámetro 14=1 del Syspar de CDS/ISIS

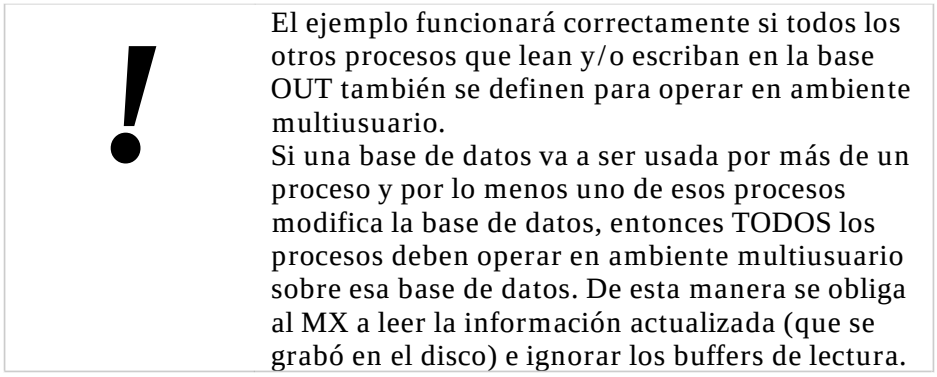
Ejemplos:

- Poner en ambiente multiusuario dos bases de datos, CDS y OUT:

```
mx mast CDS from=120 append=OUT now -all
```

Colocar en ambiente multiusuario sólo una base de datos. La base CDS estará en ambiente monousuario, en cambio la base OUT estará en ambiente multiusuario.

```
mx CDS from=120 mast append=OUT now -all
```



El ejemplo funcionará correctamente si todos los otros procesos que lean y/o escriban en la base OUT también se definen para operar en ambiente multiusuario.
Si una base de datos va a ser usada por más de un proceso y por lo menos uno de esos procesos modifica la base de datos, entonces TODOS los procesos deben operar en ambiente multiusuario sobre esa base de datos. De esta manera se obliga al MX a leer la información actualizada (que se grabó en el disco) e ignorar los buffers de lectura.

Otros Parámetros

Delimitadores

?

El signo de igual usado como delimitador puede sustituirse por el signo de interrogación ?

```
mx CDS from=10     es correcto
mx CDS from?10     es equivalente al anterior
```

Indicadores (*prompts*) predeterminados

p1|p2=<prompt>

Es posible cambiar los indicadores (*prompt*s) predeterminados del MX, durante una sesión de MX.

```
parámetro p1=<prompt1>     cambia .. a <prompt1>
parámetro p2=<prompt2>   cambia -> a <prompt2>
```

Ejemplo:

Se cambiarán los prompts de tal manera que MX entre registro y registro diga: presione <enter> para próximo registro o <x> para salir. (*prompt1*) y cuando acaben los registros para visualizar presente el mensaje: presione <x> para salir o tipee una expresión de búsqueda (*prompt2)*.

```
mx cds "p1=<enter> para próximo registro <x> para salir" "p2=presione <x>
para salir o tipee una expresión de búsqueda:" plants
```

Visualización de uno de los registros de la búsqueda (no el último):

```
mfn= 13

24 «Experience with three vapour methods for measuring water potential in
plants»

26 «^c1965»

30 «^ap. 369-384^billus.»

44 «Methodology of plant eco-physiology: proceedings of the Montpellier
Symposium»

69 «Paper on: <plant physiology><water><pressure><measurement and
instruments>»

70 «Barrs, H.D.»

70 «Slatyer, R.O.»

50 «prueb»

<enter> para proximo registro <x> para salir
```

Visualización del último registro de la búsqueda:

```
mfn= 68

24 «Some important animal pests and parasites of East Pakistan»

26 «^c1966»

30 «^ap. 285-291^billus.»

44 «Scientific problems of the humid tropical zone deltas and their
implications: proceedi+»

50 «Incl. bibl.»

69 «Paper on:
<pests><parasites><biology><ecology><plants><agriculture><public healt»

70 «Yosufzai, H.K.»

presione <x> para salir o tipee una expresión de búsqueda:
```

Parámetro *trace*

trace

El parámetro *trace* activa la salida de información interna de algunos procesos.

Parámetro *mfrl*

mfrl

El parámetro *mfrl* muestra el largo mayor de registro que se ha procesado hasta ese momento.

Ejemplo:

```
mx CDS now -all mfrl
MFN=1 -> MFRmfrl=408
MFN=3 -> MFRmfrl=450
MFN=21 -> MFRmfrl=452
MFN=27 -> MFRmfrl=822
MFN=104 -> MFRmfrl=980
```

MX: código de retorno de ejecución

MX termina la ejecución con valor de salida 1 o 0 dependiendo de si ha ocurrido o no algún error. Este código de retorno puede ser usado para el control de ejecución de procesos por lotes (*errorlevel* en el MS-DOS).

Utilitarios del archivo maestro

MXF0 - Programa

El programa MXF0 analiza todos los registros de un archivo maestro, produciendo información sobre sus campos y sobre la frecuencia de los caracteres.

El resultado de la ejecución del MXF0 es un archivo maestro con un registro que contiene la siguiente información:

- Nombre de la base de datos, fecha, hora, cantidad de registros, cantidad de registros activos, cantidad de registros borrados lógicamente y cantidad de registros borrados físicamente.
- Para cada *tag* de campo distinto hallado en los registros de entrada presenta una ocurrencia de un campo repetible que contiene: la etiqueta (*tag*), la frecuencia, total de ocurrencias, largo mayor y menor y la cantidad total de caracteres.
- Un campo repetible con una ocurrencia para cada carácter diferente hallado en los datos de entrada, su código hexadecimal y la cantidad de veces que aparece.

Este programa utilitario puede ser usado para producir una lista de etiquetas de los campos presentes en una base de datos. Sirve también para verificar si:

- campos obligatorios están presentes,
- campos no repetibles ocurren más de una vez,

- campos de longitud fija (como las fechas normalizadas) tienen el largo correcto, etc.

MXF0 - Presentación

La siguiente línea procesa el archivo maestro *cds* localizado en el directorio \cisis\cds, reinicializa el archivo maestro *x* (en el directorio actual) y almacena el resultado en el primer registro, como se indica a continuación:

```
mxf0 \cisis\cds\cds create=x 0
```

El registro resultante en la base x es

```
mfn=     1
1001  «cds»
1003  «20051014 09:44:40 Fri»
1009  «      149»
1010  «      149»
1011  «        0»
1012  «        0»
1013  «      150»
1020  «^t024^d      148^o      148^l    6^u  179^n      9589»
1020  «^t025^d        5^o        7^l    2^u   22^n        55»
1020  «^t026^d      147^o      148^l    6^u  101^n      2897»
1020  «^t030^d      146^o      146^l    6^u   36^n      2484»
1020  «^t044^d       80^o       80^l   22^u  112^n      7525»
1020  «^t050^d       99^o       99^l   11^u  209^n      2043»
1020  «^t069^d      148^o     1134^l    3^u   36^n     16318»
1020  «^t070^d      121^o      163^l    8^u   35^n      2493»
1030  «^tall^x20^n      4516»
1030  «^tall^x22^n         8»
1030  «^tall^x27^n        29»
...
1030  «^tall^xa1^n        10»
1030  «^tall^xa2^n        10»
..x
```

Este registro informa que el archivo maestro \cisis\cds\cds tiene 149 registros activos, y 8 campos diferentes de datos: etiquetas 24, 25, 26, 30, 44, 50, 69 y 70.

El campo con tag 24 (título) ocurre una vez en cada uno de los 149 registros (la ocurrencia más corta es de 6 caracteres de longitud, la más larga 179, y en conjunto estas ocurrencias suman 9589 bytes).

El campo 70 (autor) falta en 27 registros, y tiene en total 163 ocurrencias.

Considerando todos los campos de datos de \cisis\cds\cds, el cáracter código 20 (carácter de espacio en hexadecimal, en ASCII decimal: 2x16 + 0 = 32) ocurre 4.516 veces. En cambio, el carácter código *a1* (i con tilde agudo, en ASCII decimal: 10x16+1=161) en hexadecimal, sólo 10 veces.

MXF0 - Sintaxis

```
mxf0 <dbname> [create=]<dbnout> [<tnrecs>] [noedit] [tell=<n>]
```

Los parámetros deben ir en el orden indicado, por ejemplo:

```
mxf0 cds create=x noedit 0
```

Dará error, la forma correcta de ponerlo es:

```
mxf0 cds create=x 0 noedit
```

Parámetros obligatorios

<dbname> <dbnout>

Nombre del archivo maestro de entrada

<dbname>

Archivo maestro sobre el que se realizará el análisis

Nombre del archivo maestro de salida

<dbnout>

Archivo maestro de salida, es el archivo maestro que contiene la tabla generada. Si ya existe un archivo maestro con ese nombre los registros se suman.

Crear archivo maestro de salida

create=<dbnout>

El archivo maestro de salida es creado. Si existe un archivo maestro con ese nombre toda su información se pierde.

Cantidad aproximada de registros

<tnrecs>

Número de registros del archivo maestro de entrada. En caso de desconocerlo, si especificado, debe indicar cero. Cuando el parámetro es diferente de cero, el programa finaliza la ejecución con código de retorno igual a cero sólo si la cantidad de registros leídos sea igual al valor del parámetro.

Parámetros opcionales

[noedit] [tell=<n>]

Eliminación de espacios en blanco

noedit

Cuando está presente suprime todos los espacios a la izquierda en los campos de datos de salida.

Cuando este parámetro no está presente (valor por defecto), el programa utiliza longitud fija para los campos, rellenando con blancos a izquierda.

Información sobre la ejecución del proceso

tell=<n>

Envía un mensaje de estado al *stderr (standard error)* cada <*n*> registros.

Este parámetro se explica detalladamente en Apéndice Parámetros de uso general.

MXF0 - Salida

El registro de salida de MXFO tiene los campos siguientes:

TAG	CONTENIDO
1001	Nombre del archivo maestro de entrada.
1003	Fecha y hora.
1009	Cantidad de registros procesados.
1010	Cantidad de registros activos.
1011	Cantidad de registros borrados lógicamente.
1012	Cantidad de registros borrados físicamente.
1013	Siguiente MFN a ser asignado.
1020	^tTAG ^dDOCS ^oOCCS ^lMINLEN ^uMAXLEN ^nDATA BYTES
1030	^tall ^xCHRCODE ^nCHRFREQ

El campo repetible 1020 tiene una ocurrencia para cada etiqueta de campo diferente hallada en los registros de entrada:

SUBCAMPO	CONTENIDO
TAG	Etiqueta.
DOCS	Cantidad de registros que tienen TAG.
OCCS	Total de ocurrencias
MINLEN	Largo más corto.
MAXLEN	Largo más largo.
DATA BYTES	Total de caracteres

El campo repetible 1030 tiene una ocurrencia individual para cada carácter diferente hallado en la totalidad de campos de datos de la entrada:

SUBCAMPO	CONTENIDO
CHRCODE	Código del carácter, en formato hexadecimal (x00-xFF)
CHRFREQ	Frecuencia del código.

MXTB - Programa

El programa MXTB permite contar el contenido de los campos, por ejemplo, cantidad de veces que aparece cada autor, cantidad de veces que aparece cada descriptor, cantidad de veces que aparece un autor y una revista (juntos), etc.

El resultado de la ejecución del MXTB es un archivo maestro con un registro por palabra/frase diferente encontrada. Cada registro contiene, entre otros datos, cadena de caracteres, frecuencia, etc.

MXTB es un programa de tabulación en múltiples columnas para registros del archivo maestro, que define las claves de tabulación mediante lenguaje de formato.

A medida que los registros son leídos, el formato provisto es ejecutado y los datos resultantes se toman como valores para las correspondientes claves de tabulación.

Todas las n-tuplas posibles son tabuladas y, al final, se entregan como registros individuales del archivo maestro junto con sus frecuencias.

El proceso de tabulación se lleva a cabo en memoria, usando una técnica de *hash*. Esto requiere una cantidad de memoria suficiente para cargar todas las n-tuplas generadas, sus frecuencias, más un espacio adicional. Cuanto mayor sea este espacio, más rápida será la ejecución (se verá con mayor detenimiento cuando se explique el parámetro *class* más adelante en este capítulo).

Este programa utilitario puede usarse para producir tablas acumulativas de algunos datos almacenados en el archivo maestro. Por ejemplo: una lista de autores conteniendo las frecuencias que tienen en una base de datos bibliográfica, o una lista de éstos desagregados por año de publicación, u otro dato.

Optativamente, MXTB toma una especificación adicional de formato para conseguir un valor para el proceso de tabulación. En este caso, la información numérica de salida es la suma de estos valores para una n-tupla en particular.

MXTB - Presentación

```
mxtb \cisis\cds\cds create=frecaut 40:(v70/)
```

Esta línea de comando escribe un registro en el archivo maestro frecaut para cada ocurrencia diferente del campo 70 en *\cisis\cds\cds*, como se indica a continuación:

```
MFN 1
1 «Magalhaes, A.C.»
998 «999999997»
999 «2»
MFN 2
1 «Franco, C.M.»
998 «999999998»
999 «1»
...
```

Campos de los registros de frecaut:

TAG	CONTENIDO
1	Cadena de caracteres a contar, en el ejemplo, un autor generado por el formato (v70/). (categoria)
998	Máximo posible de ocurrencias de una categoría menos la frecuencia hallada. (999999999 - frecuencia)
999	Cantidad de veces que ocurre el contenido del campo con TAG 1 de frecaut en el archivo de entrada \cisis\cds\cds. (frecuencia)

El campo 998 provee una forma sencilla de realizar listados en orden descendente de las categorías halladas (por ejemplo, con el comando MSRT freqaut 10 v998).

El MXTB puede también tomar un valor externo a ser tabulado, esto quiere decir: en lugar de sumar el valor uno por cada ocurrencia hallada puede sumar un valor entrado mediante un formato, incluso este valor no tiene por qué ser fijo, puede ser un campo de la misma u otra base de datos, donde el contenido del campo 999 será la suma de los valores hallados en cada caso.

Ejemplos de uso del MXTB

Si tiene ingresados en la base de datos el costo de los documentos, fecha de adquisición, y área a la cual pertenecen. Se puede realizar una tabla que contenga –por ejemplo– el dinero invertido en documentos en el último año por cada área.

Suponiendo que entre los datos ingresados en los registros figura el costo del documento y la editorial, con MXTB fácilmente se puede realizar una tabla que contenga la editorial y el monto que se le pagó.

```
mxtb holdings x 40:(v26^b/) tab=v90
```

Para los primeros 40 caracteres de cada editor distinto almacenado como subcampo ^*b* del campo *26* en la base holdings, escribe un registro en la base x, como se indica a continuación:

```
MFN 1
1 «Unesco Press»
998 «999993999»
999 «6000»
MFN 2
1 «ESCAP»
998 «999999969»
999 «30»
MFN 3
1 «Praeger Publishers»
998 «999999599»
999 «400»
...
```

El ejemplo devuelve en el campo 999 la suma del contenido del campo v90 de cada categoría, entonces –en lugar de sumar 1 por cada elemento encontrado– suma lo que indica el campo 90. Para utilizar el parámetro *tab* el campo 90 de todos los registros debe contener un valor numérico, de lo contrario los registros cuyos campos 90 no contengan un valor numérico no se tomarán en cuenta.

MXTB - Sintaxis

```
mxtb <dbn> [create=]<dbnout> <key> [<key> [...]] [<option> [...]]
keys:    keylen:key_fmtspec

options: {from|to|loop|count|tell|btell}=<n>
         tab=<tab_val_fmt>
         class=1000
```

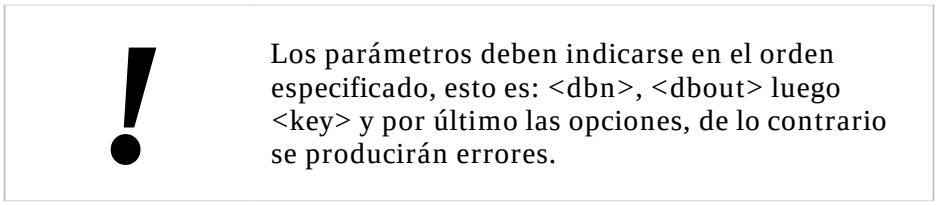
```
            bool=<expr>
            {min|max}{avg|freq}=<n> -
             uctab={<file>|ansi}

Ex: mxtb in out len1:fmt1 len2:fmt2 len3:fmt3

     out = 1  key/key1_value (max len1 chars)
           2  key/key2_value (max len2 chars)
           3  key/key3_value (max len3 chars)
         998  999999999 - key_frequency
         999  key_frequency

Ex: mxtb in out len:fmt tab=Vtag

     out = 1  key_value (max len chars)
         998  999999999 - Vtag_subtotal
         999  Vtag_subtotal
```



Los parámetros deben indicarse en el orden especificado, esto es: <dbn>, <dbout> luego <key> y por último las opciones, de lo contrario se producirán errores.

Parámetros obligatorios

<dbn> <dbnout> <key>

Nombre del archivo maestro de entrada

<dbn>

Nombre del archivo maestro del cual MXTB obtiene los datos para generar la tabla.

Nombre del archivo maestro de salida

<dbnout>

El archivo maestro de salida, es el archivo maestro que contiene la tabla generada. Si ya existe un archivo maestro con ese nombre los registros se suman.

Crear archivo maestro de salida

create=<dbnout>

El archivo maestro de salida es creado. Si existe un archivo maestro con ese nombre toda su información se pierde.

Clave

<key>

El parámetro <key> se aplica a cada una de hasta 8 claves de tabulación, y tiene la siguiente forma:

```
<longitud de la clave>:<formato que especifica la clave>
```

Longitud máxima de la clave

<keylen>

<keylen> es la cantidad máxima de caracteres retornados por la especificación de formato <key_fmtspec>. Esto significa que si la cadena de caracteres generada por <key_fmtspec> es mayor que <keylen>, se toman en cuenta sólo los primeros <keylen> caracteres.

Formato que especifica la clave

Parámetro <key_fmtspec>

Es el formato que genera las claves de tabulación. Por ejemplo, para tabular por editor el formato debe generar los editores, uno por línea.

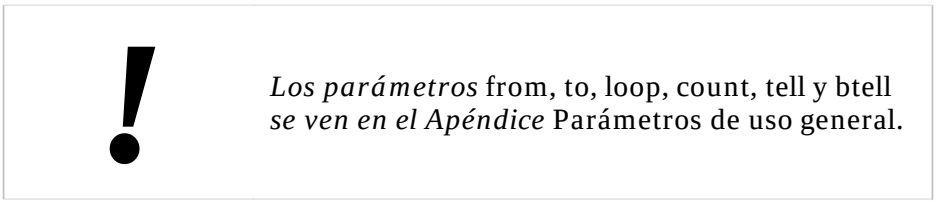
Parámetros opcionales

bool=<exp>

{from|to|loop|count|tell|btell}=<n>

tab=<tab_val_fmt>

class=<n>



Los parámetros from, to, loop, count, tell y btell *se ven en el Apéndice* Parámetros de uso general.

Procesar el resultado de una búsqueda

bool=<exp>

Se puede tabular el resultado de una búsqueda, indicando el parámetro *bool* y la expresión booleana. Esto permite, por ejemplo, contar todos libros con una determinada propiedad: todas las revistas de la editorial X o todos los libros de un área y el costo total, etc.

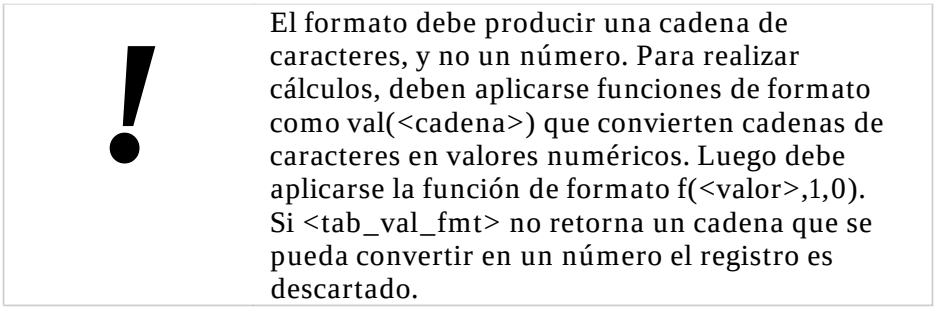
Tabulación del resultado de formato

tab=<tab_val_fmt>

El valor por defecto de tabulación (si no se especifica *tab*) es 1. Esto hace que MXTB sume el valor 1 a una categoría por cada ocurrencia hallada.

tab permite cambiar el valor por defecto mediante un formato, el cual puede especificar un valor constante (como ser '5'), o puede especificar un campo de dónde sacar el valor (por ejemplo el campo precio de una base de datos de libros).

tab permite realizar tablas que contengan, por ejemplo, dinero que la biblioteca invirtió en libros y revistas por cada área.



El formato debe producir una cadena de caracteres, y no un número. Para realizar cálculos, deben aplicarse funciones de formato como val(<cadena>) que convierten cadenas de caracteres en valores numéricos. Luego debe aplicarse la función de formato f(<valor>,1,0). Si <tab_val_fmt> no retorna un cadena que se pueda convertir en un número el registro es descartado.

Cantidad de categorias

class=<n>

Asigna espacio para hasta <n> categorías para ser tabuladas.

El valor por defecto para este parámetro es 1000.

A los efectos de *hashing*, se recomienda especificar un parámetro *class* con valor igual entre 2 y 3 veces la cantidad esperada de categorías.

Es posible visualizar la cantidad de memória disponible a través del parámetro opcional trace.

MXTB - Salida

Los registros de salida de MXTB están compuestos de los siguientes campos:

MFN n	
TAG	**CONTENIDO**
1	Valor producido por<key_fmtspec1>, hasta<len1> [characteres]
2	Valor producido por<key_fmtspec2>, hasta<len2> [characteres]
3	Valor producido por<key_fmtspec3>, hasta<len3> [characteres]
998	Valor 999999999 menos el valor del campo 999
999	Frecuencia para la categoría formada por los campos 1, 2, 3, ...

Cuando se usa la opción *tab=<tab_val_fmt>* el campo 999 contiene la suma de los valores producidos por la especificación de formato *<tab_val_fmt>* para cada registro perteneciente a esa categoría.

Los registros que no generan salida para la primera especificación <key> son descartados y, por lo tanto, no se incluyen en la tabulación.

MXCP - Presentación

El comando:

```
mxcp \cisis\cds\cds create=x clean undelete
```

Reinicializa el archivo maestro **x** (en el directorio actual) y le copia todos los registros del archivo maestro cds localizado en el directorio \cisis\cds.

Ningún campo de datos en el archivo maestro **x** tendrá espacios en blanco al comienzo o al final del campo debido a que se especificó la opción *clean*. Es más, cada carácter no imprimible que aparezca en los campos de entrada será convertido en un carácter de espacio en la salida.

Debido a que se especificó la opción *undelete*, todos los registros borrados lógicamente del archivo maestro \cisis\cds\cds estarán activos en el archivo maestro x.

MXCP puede convertir en repetibles campos que no lo son, siempre y cuando haya un carácter separador de los datos.

En el siguiente ejemplo MXCP convierte en repetible el campo con tag 70 conteniendo el signo "**;**" como delimitador de repetición de campo.

```
mxcp in create=out repeat=;,70
```

Para el ejemplo suponemos que los registros 1 y 2 de la base in tienen en el campo con tag 70:

```
MFN 1
70 «Magalhaes, A.C.; Franco, C.M.»
MFN 2
70 «Bosian, G.»
```

Entonces en out (el archivo maestro resultante) los registros quedan de esta forma:

```
MFN 1
70 «Magalhaes, A.C.»
```

```
70 «Franco, C.M.»
MFN 2
70 «Bosian G.»
```

Cuando se usa el parámetro *repeat*, MXCP también realiza su proceso de "limpieza".

Si no se especifica etiqueta en el parámetro *repeat*, todos los campos presentes serán incluídos:

```
mxcp in create=out repeat=%
```

Puede también especificarse una lista de campos:

```
mxcp in create=out repeat=%,70,99,100,200
```

Un rango:

```
mxcp in create=out repeat=%,99/105
```

Una mezcla de ambos:

```
mxcp in create=out repeat=%,70,99/105,110
```

Otra característica de MXCP es la de suprimir los signos de puntuación final. El ejemplo siguiente remueve el carácter de punto final en el campo 70:

```
mxcp in create=out period=.,70
```

Para el ejemplo suponemos que los registros 1 y 2 de la base in tienen en el campo con tag 70:

```
MFN 1
70 «Magalhaes, A.C.»
70 «Franco, C.M.»
MFN 2
«Bosian G.»
«xxx.»
```

Entonces en out (el archivo maestro resultante) los registros quedan así:

```
MFN 1
70 «Magalhaes, A.C»
70 «Franco, C.M»
MFN 2
«Bosian G»
```

```
«xxx.»
```

MXCP también realiza un procedimiento de cambio de patrones, permitiendo que los contenidos de los campos de entrada especificados se cambien a medida que son leídos.

Tipeando exactamente lo que se presenta a continuación (finalizando cada línea con <enter>) se crea una tabla de conversión para reemplazar los corchetes angulares por el signo por ciento (%):

```
mx seq=con create=xtable -all now
<|
>|
><|%
> <|%
> <|%
Paper on: <|
<Ctrl>Z
```

Entonces los campos con etiqueta 69 de la base cds cuyo contenido es:

```
MFN 1
69 «Paper on: <plants><water><plant transpiration>»
MFN 2
69 «Paper on: <plant physiology><plant transpiration><measurement and
instruments»
```

Al ejecutar:

```
mxcp cds create=out gizmo=xtable repeat=%,69
```

Provoca que los registros con etiqueta 69 de cds se transformen en out de la siguiente manera:

```
MFN 1
69 «plants»
69 «water»
69 «plant transpiration»
MFN 2
69 «plant physiology»
69 «plant transpiration»
69 «measurement and instruments»
```

MXCP - Sintaxis

```
mxcp {in=<file>|<dbin>} [create=]<dbout> [<option> [...]]
options: {from|to|loop|count|tell|offset}=<n>
         gizmo=<dbgiz>[,tag_list>]
         undelete
         clean [mintag=1] [maxtag=9999]
         period=.[,<tag_list>]
         repeat=%[,<tag_list>]
         log=<filename>

 Ex: mxcp in create=out clean period=.,3 repeat=;,7

      in = 3 «   Field 3 occ 1. »
           3 «Field 3 occ 2 . »
           7 « Field 7/1;Field 7/2 ;Field 7/3.»
     out = 3 «Field 3 occ 1»
           3 «Field 3 occ 2»
           7 «Field 7/1»
           7 «Field 7/2»
           7 «Field 7/3.»
```

Los parámetros presentes deben estar en el orden estipulado, primero <dbin> luego <dbout> y por último las opciones, de lo contrario se generará un error.

Nombre del archivo maestro de entrada

<dbin>

Archivo maestro de entrada.

Nombre del archivo maestro de salida

<dbout>

En el archivo maestro de salida es donde se copiarán los registros provenientes de *dbin*.

Si *dbout* no existe se producirá un error; para crearlo se debe utilizar el parámetro *create,* que se explica a continuación.

Crear archivo maestro de salida

create=<dbnout>

Cuando el archivo maestro de salida no existe se puede crear mediante este parámetro.

Si *dbout* existe se perderán todos sus datos, ya que la base de datos de salida es reinicializada antes de copiar los registros.

Parámetros opcionales [option]

{from|to|loop|count|tell|offset}=<n>

undelete

gizmo=<dbgiz>[,tag_list>]

period=.[,<tag_list>]

repeat=%[,<tag_list>]

clean [mintag=1] [maxtag=9999]

log=<filename>

Los parámetros *from*, *to*, *loop*, *count*, *tell* y *offset* se ven en el *Apéndice I - Parámetros de uso general.*

Recuperar registros borrados

undelete

Recupera registros del archivo maestro (activa los registros de la entrada que estén lógicamente borrados).

```
mxcp cds newcds undelete
```

Cambio global de patrones

gizmo=<dbgiz>[,<tag_list>]

Aplica un procedimiento gizmo a los datos de entrada, usando el archivo maestro *dbgiz* como tabla gizmo; puede aplicarse a todos los campos de un registro o sólo a los campos especificados en <tag_list>.

La lista <tag_list> puede especificarse de las siguientes maneras:

- Una etiqueta.
- Una lista de etiquetas separadas por coma.
- Un rango de etiquetas en la forma *valor inferior / valor superior*.
- Una combinación estas opciones.

Si se especifica más de un gizmo=<dbgiz>[,<tag_list>], el segundo se ejecuta sobre el registro que resulta del procedimiento del primer gizmo, y así sucesivamente.

```
mxcp cds newcds gizmo=xtable
mxcp cds newcds gizmo=xtable,24,50/70
```

El parámetro gizmo se desarrolla ampliamente en el *Apéndice I - Parámetros de uso general*.

Supresión de caracteres de puntuación

```
period=<char>[,<tag_list>]
```

MXCP provee la posibilidad de suprimir el carácter <*char*> como signo de puntuación final; puede ser aplicado a todos los campos de un registro o sólo a los campos presentes en <tag_list>.

La lista <tag_list> puede especificarse de las siguientes maneras:

- Una etiqueta
- Una lista de etiquetas separadas por coma
- Un rango de etiquetas en la forma *valor inferior / valor superior*.
- Una combinación de estas opciones.

```
mxcp cds newcds period=;
mxcp cds newcds period=;,24,50/60,70
```

Convertir campos en repetibles

repeat=<char>[,<tag_list>]

Convierte en repetible los campos de entrada que tienen <*char*> como delimitador; puede ser aplicado a todos los campos de un registro o sólo a los campos presentes en <tag_list>.

La lista <tag_list> puede especificarse de las siguientes maneras:

Una etiqueta.

- Una lista de etiquetas separadas por coma.
- Un rango de etiquetas en la forma *valor inferior* / *valor superior*.
- Una combinación de estas opciones.

```
mxcp cds newcds repeat=/
mxcp cds newcds repeat=;,24,50/60,70
```

Supresión de espacios en blanco

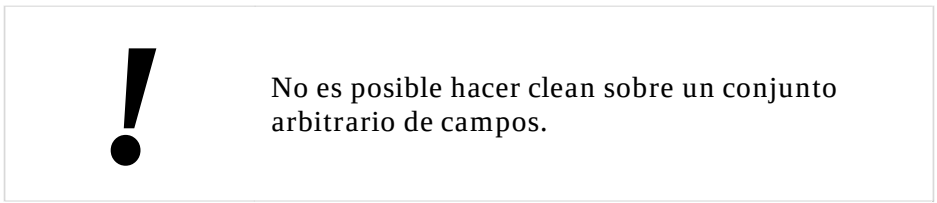
clean

Suprime todos los espacios en blanco al comienzo y al final del campo y reemplaza todos los caracteres no imprimibles por espacios en blanco.

Si se usa *repeat* o *period* la opción *clean* se inicia automáticamente.

La opción *clean* se aplica a todos los campos del registro.

```
mxcp cds newcds clean
```



Eliminación de campos por tag

mintag=<n>

Borra todas las ocurrencias de los campos con tag menor que <n>. Por defecto n=1

```
mxcp cds newcds mintag=10
 maxtag=<n>
```

Borra todas las ocurrencias de los campos con tag mayor que <n>. Por defecto n=9999

```
mxcp cds newcds maxtag=70
```

Registro de eventos

log=<filename>

Direcciona los mensajes de control y estado al archivo <*filename*>.

MXCP - Salida

MXCP no produce salidas de registros vacíos. Los registros de entrada o procesados que no tengan campos son descartados. Los registros borrados lógicamente son procesados sólo si se usa la opción *undelete*.

Puede usarse el mismo archivo maestro tanto para la entrada como para la salida, aunque sólo será posible reorganizar un archivo maestro copiándolo a uno nuevo. La opción *create*=<*dbout*> asegura que <*dbout*> será creado o reinicializado.

El comando:

```
mxcp in create=out period=.,3 repeat=;,7
```

Para el archivo maestro *in*:

```
MFN 1
3 «Field 3 occ 1. »
3 «Field 3 occ 2 .»
7 «Field 7/1; Field 7/2; Field 7/3.»
MFN 2
...
```

Crea *out* con:

```
MFN 1
3 «Field 3 occ 1»
```

```
3 «Field 3 occ 2»
7 «Field 7/1»
7 «Field 7/2»
7 «Field 7/3»
MFN 2
...
```

Produce los mensajes siguientes:

```
*** mfn 1 tag=3/1 -> rejected char
*** mfn 1 tag=3/1 -> rejected char
*** mfn 1 tag=3/1 . -> rejected char
*** mfn 1 tag=3/2 -> rejected char
*** mfn 1 tag=3/2 . -> rejected char
*** mfn 1 tag=3/2 -> rejected char
*** mfn 1 tag=7/1 -> rejected char
```

MSRT - Programa

El programa MSRT ordena un archivo maestro, en forma ascendente, siguiendo las claves de ordenación generadas por una especificación de formato dada.

Después que un archivo maestro es ordenado, el registro con mfn=1 contiene la clave de ordenación más baja, el registro con mfn=2 la segunda clave de ordenación más baja, y así sucesivamente.

MSRT - Presentación

El comando:

```
msrt \cisis\cds\cds 60 mhu,v24
```

Ordena el archivo maestro cds localizado en el directorio \cisis\cds según los primeros *60* caracteres del campo *24* (transformado apropiadamente a mayúsculas).

El comando:

```
msrt mxtb_out 9 v998
```

Ordena el archivo maestro mxtb_out según los valores almacenados en el campo 998 (presumiendo que es de longitud fija y con ceros a la izquierda).

Mientras que el comando:

```
msrt mxtb_out 9 f(999999999-val(v999),9,0)
```

Ordena el archivo maestro mxtb_out en orden descendente, de acuerdo al valor numérico almacenado en el campo 999.

MSRT - Sintaxis

```
msrt <dbname> <keylen> <keyfmt> [-mfn] [tell=<n>]
```

Los parámetros presentes deben estar en el orden estipulado, esto es <dbname>, <keylen>, luego <keyfmt> y, por último, las opciones, de lo contrario se producirá un error.

Parámetros obligatorios:

<dbname> <keylen> <keyfmt>

Nombre del archivo maestro de entrada

<dbname>

Archivo maestro a ser ordenado en su mismo master.

Longitud máxima de la clave

<keylen>

Cantidad máxima de caracteres a comparar.

Generación clave

<keyfmt> | tag=<n>

Formato que aplicado a los registros genera las claves de ordenación.

Para especificar un campo de datos como clave de ordenación, sin la necesidade de ejecutar un formato, el parámetro <keyfmt> debe ser **tag=<n>**.

Parámetros opcionales

-mfn

+del

tell=<n>

El parámetro tell se explica detalladamente en el Apéndice Parámetros de uso general.

Mantener MFNs originales

-mfn

Conserva los *MFN* originales. Por defecto, los registros son renumerados, despues de ordenados.

Borrar claves iguales

+del

Borra las claves iguales de registros consecutivos, despues de ordenados.

MSRT - Salida

Cuando se ordena un archivo maestro, el programa MSRT actualiza solamente el .xrf, intercambiando las entradas correspondientes de cada par de registros que están siendo ordenados.

Al final de este proceso, a menos que se use la opción -mfn, se actualiza el archivo .mst, asignando los números de registros al campo mfn del leader del registro.

Esta actualización se realiza en el mismo lugar de los registros, lo que significa que los registros se modifican en sus propias ubicaciones, por tanto, la organización del archivo maestro procesado no cambia. De esta manera se ahorra el tiempo que se invierte en grabar todos los registros ordenados en el archivo maestro.

RETAG - Programa

El programa *RETAG* cambia el tag de los campos del archivo maestro, según una tabla de reenumeración dada. También puede realizar un desbloqueo (*unlock*) del archivo maestro.

Las operaciones que realiza RETAG se llevan a cabo sobre el mismo registro, lo que significa que los registros son modificados en las ubicaciones originales y, por lo tanto, la organización del archivo maestro procesado no cambia.

RETAG - Presentación

El comando:

```
retag \cisis\cds\cds xtable
```

Procesa todos los registros activos del archivo maestro cds localizado en el directorio \cisis\cds, reescribiendo el segmento del directorio de los registros correspondientes acorde a una tabla xtable de renumeración de tag localizada en el directorio actual.

Una tabla de renumeración es un archivo secuencial que tiene una línea por cada tag a ser renumerado, como se indica a continuación:

```
24 240
70 700
69 690
```

El comando anterior cambia el tag 24 a 240, el tag 70 a 700, y el tag 69 a 690.

En este ejemplo, sólo son rescritos los registros que contienen, por lo menos, una ocurrencia de los tag 24, 70 o 69.

Además, el programa *RETAG* puede ser usado para dejar sin efecto los bloqueos del archivo maestro.

El comando:

```
retag \cisis\cds\cds unlock
```

En lugar de especificar una tabla de renumeración de etiquetas, especifica la palabra clave unlock, que deja sin efecto los *exclusive write lock* o *data entry locks* y todos los *record locks* existentes en el archivo maestro cds localizado en el directorio \cisis\cds.

RETAG - Sintaxis

```
retag <dbname> {<retag.tab>|unlock} [<option> [...]]
```

Parámetros obligatorios

<dbname> <retag.tab>

Archivo maestro de entrada

<dbname>

Archivo maestro a ser reetiquetado o desbloqueado (*unlocked*).

Tabla de renumeración

<retag.tab>

Nombre de un archivo secuencial que provee una tabla de reetiquetado, o la palabra clave *unlock*.

Si <retag.tab> es *unlock*, esto indica al programa que debe realizarse un desbloqueo del archivo maestro.

Una tabla de reetiquetado tiene el formato:

```
<tag> <new tag>
```

Puede contener hasta 5.461 entradas en la version standard.

Parámetros opcionales

from=<n>

to=<n>

tell=<n>

shift=<n>

Los parámetros from, to, shift y tell *se ven en el Apéndice* Parámetros de uso general.

RETAG - Salida

Las operaciones RETAG son realizadas sobre el mismo registro físico (*in-place*), sin importar si lo solicitado es un reetiquetado o un desbloqueo.

En la operación de reetiquetado, sólo se reescribe el segmento del directorio de los registros procesados.

En la operación de desbloqueo, sólo se reescribe el registro de control del archivo maestro y el segmento *leader* de los registros procesados.

CTLMFN - Programa

El programa CTLMFN despliega y actualiza el registro de control de un archivo maestro.

Debe usarse antes de ejecutar el programa MKXRF para restaurar todos los registros activos en un archivo maestro reinicializado lógicamente, para establecer el número siguiente a ser asignado en el archivo maestro y definir la cantidad de datos a ser analizados.

CTLMFN - Presentación

El comando:

```
ctlmfn \cisis\bases\cds
```

Lee el registro de control del archivo maestro \cisis\bases\cds.mst y permite que sean actualizados cada uno de los campos de este registro. Estos campos son:

CAMPO	CONTENIDO
nxtmfn	MFN a ser asignado al próximo registro a crearse en la base de datos.
nxtmfb	Último número de bloque asignado al maestro (el primer bloque es 1).
nxtmfp	Siguiente posición disponible en el último bloque del archivo maestro.
mftype	Para base de datos del usuario (1 para archivos de mensajes del sistema).
reccnt	Reservado.
mfcxx1	Reservado.
mfcxx2	Cantidad de usuarios en proceso de entrada de datos (*Data Entry Lock*).
mfcxx3	*Exclusive Write Lock*.

Si se escribiera un valor inválido para *nxtmfb* se informará el número total de bloques del archivo maestro.

Los comandos

```
ctlmfn \cisis\bases\cds
```

y

```
mkxrf \cisis\bases\cds
```

permiten cambiar uno o más campos de control en \cisis\bases\cds.mst (registros de datos) y entonces crear, reinicializar y escribir el archivo de referencias cruzadas \cisis\bases\cds.xrf.

CTLMFN - Sintaxis

ctlmfn <dbname> [ok]

Nombre del archivo maestro de entrada

<dbname>

Parámetro obligatorio que indica el archivo maestro a ser procesado.

Prompt de confirmación

ok

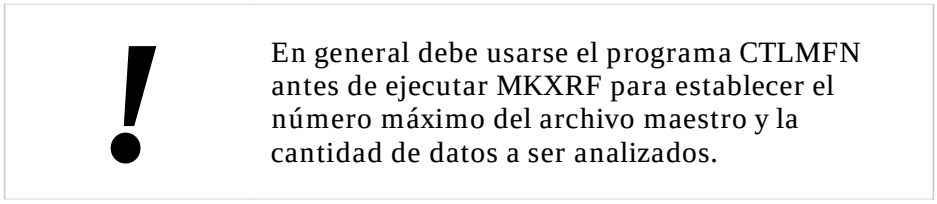
CTLMFN - Salida

El programa CTLMFN escribe solamente los primeros bytes (los bytes del registro de control) de un archivo .mst. No se cambian los restantes bytes del primer bloque del archivo maestro.

MKXRF - Programa

El programa MKXRF lee un archivo .mst y crea su archivo correspondiente .xrf.

Además puede usarse para restaurar todos los registros activos en un archivo maestro reinicializado lógicamente.



MKXRF - Presentación

El comando:

```
mkxrf \cisis\bases\cds
```

Crea, reinicializa y escribe el archivo de referencias cruzadas \cisis\bases\cds.xrf correspondiente al archivo maestro \cisis\bases\cds.mst.

El programa MKXRF puede restablecer un archivo maestro reinicializado por accidente, siempre y cuando su registro de control tenga los siguientes campos de control ajustados en forma apropiada:

CAMPO	CONTENIDO
nxtmfn	MFN a ser asignado al próximo registro a crearse en la base de datos.
nxtmfb	Último número de bloque asignado al maestro (el primer bloque es 1).
nxtmfp	Siguiente posición disponible en el último bloque del archivo maestro.

Estos valores pueden establecerse usando el programa CTLMFN, como se muestra en los siguientes comandos:

```
ctlmfn \cisis\bases\cds
mkxrf \cisis\bases\cds
```

Estos comandos permiten cambiar uno o más campos de control en el archivo \cisis\bases\cds.mst (registros de datos) para crear, reinicializar, y escribir el archivo de referencias cruzadas \cisis\bases\cds.xrf.

Descripción de la ejecución de CTLMFN en el ejemplo:

a) Si *nxtmfn* es desconocido, deberá establecerse el máximo mfn *posible* (ver explicación más abajo); el *nxtmfn* real podrá ser obtenido posteriormente ejecutando el programa *MXF0* (campo 1013)
b) *nxtmfb* es la cantidad de bloques del archivo maestro \cisis\bases\cds.mst que están para ser analizados; si se escribiera un valor inválido entonces se informará el número total de bloques.

c) *nxtmfp* es la última posición en el bloque *nxtmfb* del archivo maestro que está para ser analizado; un valor de 512 fuerza a *MKXRF* a procesar todos los contenidos del bloque nxtmfb del archivo maestro.

MKXRF - Sintaxis

```
mkxrf <dbname>
```

Archivo maestro de entrada

<dbname>

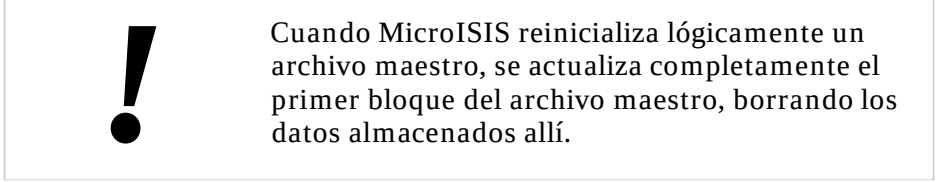
Nombre del archivo maestro a ser procesado (parámetro obligatorio).

MKXRF - Salida

El programa MKXRF lee el registro de control y entonces examina el archivo de entrada .mst hasta el lugar *nxtmfb* y *nxtmfp*; en este proceso identifica a un registro del archivo maestro si las siguientes condiciones se aplican a los componentes del *leader*:

a) MFRmfn está en el rango entre 1 y (nxtmfn-1).
b) MFRmfrl está en el rango LEADER a MAXMFRL.
c) MFRbase es hasta MFRmfrl.
d) MFRmfbwb apunta a alguna dirección previa válida.
e) MFRmfbwp está en el rango entre 0 y MSBSIZ.
f) MFRnvf es cero o positivo.
g) MFRstatus es ACTIVE o DELETED.
h) MFRbase es igual a LEADER + MFRnvf*sizeof(DIRSTRU).

La ubicación de cada registro del archivo maestro identificado se escribe en el archivo .xrf que es creado, e incluye las versiones viejas de registros.



Restaurando una base de datos dañada

A continuación se detallan procedimientos para restaurar una base de datos según sus posibles estados de problemas:

Estado			Acción
MST	**XRF**	**Información**	
OK	Dañado	MaxMFN desconocido	Usar MKXRF.
Reinicializado	OK	Tamaño del XRF	Estimar maxMFN y usar CTLMFN.
Bloqueado	OK		Usar CTLMFN, escribir ceros en los campos mfcxx2, mfcxx3

Cálculo aproximado del MaxMFN

Una estimación del valor máximo de registros (*maxMFN*) se obtiene de:

```
(tamaño_en_bytes_de_XRF/512) * 127
```

Si el valor de *maxMFN* que se establece es mayor que el número real de registros, no se generarán registros nulos ni en el .mst ni en el .xrf. El archivo .xrf se graba solamente para los registros que puedan leerse en el .mst. La base podrá ser leída sin problemas pero no podrá ser actualizada (generará el error *fatal: recxref/read*). El *maxMFN* deberá ser corregido siguiendo alguna de estas dos formas:

Tomar el valor "x recs" que se indica al final de la ejecución de *MKXRF* y luego almacenar x + 1 en el *nxtmfn* a través del *CTLMFN*.

Crear otro archivo maestro (con *create*, no con *copy*) a partir del que se tiene, a los efectos de corregir el registro de control en el nuevo archivo maestro.

ID2I - Programa

El programa I2ID recibe un archivo ASCII, con determinada estructura y genera un archivo maestro.

ID2I - Presentación

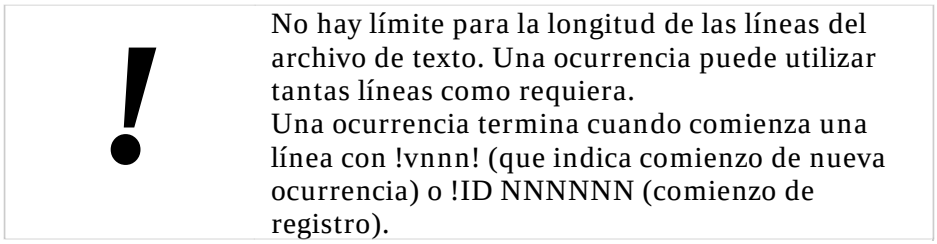
El programa recibe un archivo ASCII y devuelve un archivo maestro.

Estructura del archivo ASCII:

- !ID nnnnnn Marca de comienzo de registro con mfn=nnnnnn
- !vnnn Marca de comienzo de una ocurrencia del campo con tag nnn.

El archivo tendría la forma:

```
!ID nnnnnn
!vXXX!...contenido del tag XXX.............
!vYYY!...contenido del tag YYY.............
...
!ID nnnnnj
!vXXQ!...contenido del tag XXQ.............
!vYYQ!...contenido del tag YYQ.............
...
```



Ejemplo:

```
id2i x.txt create=newcds
```

La idea es utilizar el *ID2I* junto con el *I2ID*, que toma un archivo maestro y devuelve un archivo de texto (editable y modificable).

Mediante el *ID2I* se convierte el archivo de texto modificado en un archivo maestro:

```
I2ID cds >x.txt
```

Edit x.txt (Edita el contenido permitiendo crear, modificar y borrar registros y ocurrencias)

```
ID2I x.txt create=cds
```

ID2I - Sintaxis

```
id2i <filein> [create[/app]=]<dbout> [option [option] ... ]
options: {from|to|loop|count|offset|tell|id}=<n>
```

Parámetros obligatorios

<filein> <dbout>

Archivo ASCII de entrada

<filein>

Es un archivo ASCII con la estructura vista en la presentación.

Nombre del archivo maestro de salida

<dbout>

Archivo maestro de salida, donde se copiarán los registros provenientes de *filein*.

Si *dbout* no existe se producirá un error; para crearlo debe utilizarse *create*. Si *dbout* existe, se perderán todos sus datos, ya que es reinicializada antes de copiar los registros.

Crear archivo maestro de salida

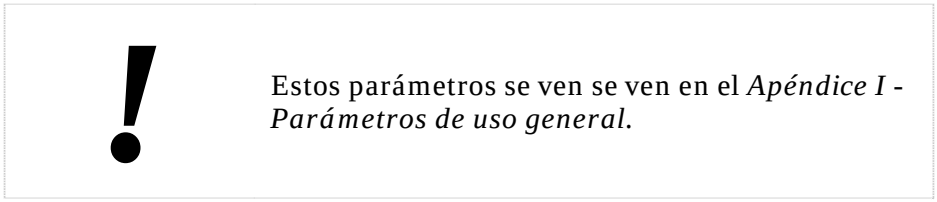
create/app=<dbnout>

Crea el archivo maestro de salida; si *dbout* existe, se perderán todos sus datos, ya que es reinicializada antes de copiar los registros. La opción **/app** significa que la numeracion se atribuye secuencialmente

Además no considera la información de MFN en las marcas de comienzo de registros (!ID nnnnnn) y accepta marcas !ID 000000.

Parámetros opcionales

{from|to|loop|count|tell|offset|id}=<n>



Estos parámetros se ven se ven en el *Apéndice I - Parámetros de uso general.*

I2ID - Programa

El programa I2ID recibe un archivo maestro y generá un archivo de texto.

I2ID - Presentación

El programa recibe un archivo maestro y devuelve un archivo de texto con la siguiente estructura:

```
!ID nnnnnn      Marca de comienzo de registro con mfn=nnnnnn
!vnnn           Marca de comienzo de una ocurrencia del campo con tag nnn.
```

Archivo de texto devuelto por *I2ID*:

```
!ID nnnnnn
!vXXX!...contenido del tag XXX.............
!vYYY!...contenido del tag YYY.............
...
!ID nnnnnj
!vXXQ!...contenido del tag XXQ.............
!vYYQ!...contenido del tag YYQ.............
...
```

No hay límite para la longitud de las líneas del archivo de texto. Una ocurrencia puede utilizar tantas líneas como requiera.

Una ocurrencia termina cuando comienza una línea con !vnnn! (indica comienzo de nueva ocurrencia) o !ID NNNNN (comienzo de registro).

Ejemplo:

```
I2ID cds >x.txt
```

La idea es utilizar el *I2ID* junto con el *ID2I*. El primero devuelve un archivo de texto (que puede ser editado y modificado) y luego, por intermedio del *ID2I*, vuelve a convertir el archivo de texto en un archivo maestro.

```
I2ID cds >x.txt
```

`edit x.txt` (edita contenido, se pueden crear, modificar y borrar registros y ocurrencias)

```
ID2I x.txt create=cds
```

I2ID - Sintaxis

```
i2id <dbn> [option [option] ... ]
options: {from|to|loop|count|offset|tell}=<n>
          1record=<tag>=<value>
```

Parámetros obligatorios:

<dbn>

Archivo maestro de entrada

<dbn>

Archivo maestro a ser convertido.

Parámetros opcionales

{from|to|loop|count|offset|tell}=<n>

Estos parámetros se ven en el Apéndice Parámetros de uso general.

CRUNCHMF - Sintaxis

```
crunchmf <dbn> <target_dbn> [<option> [...]]
```

Convierte el archivo maestro (.mst y .xrf) de un sistema operativo a otro, debido que la estructura en binario no es compatible para los diferentes sistemas. El programa detecta automáticamente el sistema en el que está operando.

options:

```
{from|to|loop|count|tell}=<n>
target={pc|linux|hpux|sun|alpha|vax|unisys|mpe|cdc|same} default: linux
format={isis|cisisX}                                     default: isis
mstxl={0|1|2|4}                                          default: as <dbn>
```

Utilitarios del archivo invertido

IFKEYS - Programa

El programa IFKEYS despliega los términos del archivo invertido y los correspondientes totales de *postings*, opcionalmente desagregados por las etiquetas de los que fueron extraídos.

IFKEYS acepta un rango de términos como parámetro y puede usarse para desplegar un conjunto de términos del archivo invertido.

IFKEYS - Presentación

El comando:

```
ifkeys \cisis\bases\cds from=plant to=plants
```

Lee el archivo invertido cds localizado en el directorio \cisis\bases comenzando en el término *plant* hasta el término plants y despliega estos términos precedidos por sus números de *postings*:

```
8|PLANT
```

```
4|PLANT ECOLOGY
1|PLANT EVAPOTRANSPIRATION
1|PLANT PHOTOSYNTHESIS
20|PLANT PHYSIOLOGY
6|PLANT TRANSPIRATION
8|PLANTS
```

Mientras que el comando:

```
ifkeys \cisis\bases\cds from=plant to=plants +tags
```

Produce la misma información y, adicionalmente, (a) incluye los tag (*Field ID*) de donde fueron generados estos términos y, (b) genera una línea diferente para cada combinación de término y tag, como se indica a continuación:

```
8| 24|PLANT
4| 69|PLANT ECOLOGY
1| 69|PLANT EVAPOTRANSPIRATION
1| 69|PLANT PHOTOSYNTHESIS
20| 69|PLANT PHYSIOLOGY
6| 69|PLANT TRANSPIRATION
6| 24|PLANTS
2| 69|PLANTS
```

Los parámetros from to no distinguen mayúsculas de minúsculas, from=plant producirá la misma salida que from=PLANT.

IFKEYS - Sintaxis

```
ifkeys <dbname> [from=<key>] [to=<key>] [+tags] [tell=<n>]
```

Archivo invertido de entrada

<dbname>

Archivo invertido a ser procesado.

Parámetros opcionales

from=<term>

to=<term>

+tags

tell=<n>

El parámetro tell se explica detalladamente en el Apéndice Parámetros de uso general.

Primer término a ser listado

from=<term>

Comienza el listado a partir del término <term>.

Ultimo término a ser listado

to=<term>

Termina el listado en el término <term>.

Mostrar información sobre etiquetas (tags)

+tags

Suma al listado el tag del cual fue extraído el término.

IFKEYS - Salida

Si se usa la opción *from=<term>* y el término especificado de comienzo no existiera, el listado comenzará en el siguiente término del archivo invertido. Si se emplea la opción *to=<term>* y el término final especificado no existiera, el listado parará en el término previo del archivo invertido.

La salida de IFKEYS puede redireccionarse a un archivo para ser procesado posteriormente por el programa MX. Por ejemplo, los comandos:

```
ifkeys cds +tags >x
mx seq=x "pft=if val(v1)=1 and val(v2)=24 then v3/ fi" now
```

Despliega las palabras del título (campo 24 en la base *cds*) que ocurren exactamente una vez.

IFLOAD - Programa

El programa IFLOAD carga un archivo invertido usando archivos de ligas ordenados según las opciones de procesamiento. Se aceptan otros formatos, además del formato de archivo de ligas estándar de MicroISIS.

También permite que sea creado sólo el archivo invertido del diccionario.

El programa IFLOAD puede usarse también para crear y reinicializar un archivo invertido.

IFLOAD - Presentación

El comando:

```
ifload \cisis\bases\cds \isis\work\cds.lk1 \isis\work\cds.lk2 tell=99
```

Carga el archivo invertido cds localizado en el directorio \cisis\bases usando los archivos de ligas de claves cortas y claves largas cds.lk1 y cds.lk2, localizados en el directorio \isis\work.

La opción *tell=99* produce un mensaje progresivo cada 99 registros de ligas procesados, mostrando la clave actual que está siendo cargada.

Se supone que los archivos de ligas anteriores están en formato de archivo de ligas estándar de MicroISIS, que tienen la siguiente presentación:

```
102 24 1 1 ABOUT
42 24 1 9 ABSENCE
```

```
6 24 1 10 ABSORPTION
87 24 1 5 ACCOUNT
136 69 1 1 ACCOUNTING
40 24 1 6 ACID
101 24 1 5 ACTION
49 24 1 6 ACTIVITIES
130 24 1 7 ACTIVITIES
23 24 1 5 ACTUAL
```

Son un conjunto de registros de longitud variable, que contienen campos que identifican el origen de la clave y la clave misma.

Los primeros 4 campos son los componentes del *posting*.

Campo	Contenido
MFN	Número de registro (master file record number).
TAG	Identificador de campo, asignado por la FST (field identifier).
OCC	Número de ocurrencia del campo
CNT	Número secuencial del término en el campo

Para permitir el uso de programas de ordenación (*sort*) estándar, IFLOAD acepta archivos de ligas con registros de longitud fija, como se indica a continuación:

```
ABOUT       102  24  1  1
ABSENCE     42   24  1  9
ABSORPTION  6    24  1  10
ACCOUNT     87   24  1  5
ACCOUNTING  136  69  1  1
ACID        40   24  1  6
ACTION      101  24  1  5
ACTIVITIES  49   24  1  6
ACTIVITIES  130  24  1  7
ACTUAL      23   24  1  5
```

El comando siguiente carga el archivo invertido cds localizado en el directorio \cisis\bases usando los archivos de ligas de longitud fija cds.lk1 y cds.lk2 localizados en el directorio \isis\work:

```
ifload \cisis\bases\cds \isis\work\cds.lk1 \isis\work\cds.lk2 +fix
```

Un procedimiento para generar un archivo invertido **x** usando la Tabla de Selección de Campos (FST) por defecto y el archivo *Stopword* es:

```
mx x fst=@ stw=@ ln1=x.ln1 ln2=x.ln2 +fix tell=100
```

```
del *.$$$
mys 37 x.ln1 x.lk1
del x.ln1
del *.$$$
mys 57 x.ln2 x.lk2
del x.ln2
ifload x x.lk1 x.lk2 +fix tell=1000
del x.lk1
del x.lk2
```

Cuando se crean varios archivos invertidos para un archivo maestro dado, y no es necesaria la operación de búsqueda por proximidad, sólo es necesario el componente MFN del *posting*. Los archivos de ligas podrían estar en el formato:

ABOUT	102
ABSENCE	42
ABSORPTION	6
ACCOUNT	87
ACCOUNTING	136
ACID	40
ACTION	101
ACTIVITIES	49
ACTIVITIES	130
ACTUAL	23

Y cargados con el comando:

```
ifload \cisis\bases\cds \isis\work\cds.lk1 \isis\work\cds.lk2 +fix/m
```

El programa IFLOAD permite cargar solamente el diccionario del archivo invertido, usando la opción *-posts*:

```
ifload authority x.lk1 x.lk2 -posts
```

Después que se carga un archivo invertido, por defecto, la señal de *I/F update is pending* se reinicializa en todos los registros asociados del archivo maestro.

Cuando varios archivos invertidos se crean para un archivo maestro dado, debe especificarse que la señal de *I/F update is pending* debe mantenerse, como se indica a continuación:

```
ifload au au.lk1 au.lk2 reset=0
ifload ti ti.lk1 ti.lk2 reset=0
```

Si debiera reinicializarse la señal de *I/F update is pending* y el nombre del archivo invertido a ser cargado fuera diferente del archivo maestro asociado, se usará la opción *master*

```
Ifload kw kw.lk1 kw.lk2 master=\cisis\bases\cds
```

IFLOAD - Sintaxis

```
ifload <dbname> {<file_lk1>|void} {<file_lk2>|void} [<option> [...]]
```

Parámetros obligatorios

<dbname>

<file_lk1>

<file_lk2>

Archivo invertido de entrada

<dbname>

Archivo invertido a ser cargado el archivo invertido <*dbname*> es creado y reinicializado incluso si éste ya existe.

Archivo de ligas de claves cortas

<file_lk1|void>

Archivo de ligas de claves cortas, ordenado.

Archivo de ligas de claves largas

<file_lk2|void>

Archivo de ligas de claves largas, ordenado.

Parámetros obligatorios

master=<name>

- {reset | posts | balan}

tell=<n>

+fix[/m]

Reinicializar marca de actualización pendiente

master=<mst_name>

Reinicializa la señal de *I/F update is pending* en el archivo maestro <*mst_name*>; por defecto se procesa al archivo maestro <*dbname*>.

Actualización pendiente

-reset

No reinicializa la señal de *I/F update is pending* en el archivo maestro <dbname> o <*mst_name*>.

No balancear diccionario

-balan

No rebalancea el diccionario (los arboles B*)

No Cargar postings

-posts

Carga solamente el diccionario. No carga los *postings* en el archivo *.ifp*.

Información sobre la ejecución del proceso

tell=<n>

Produce un mensaje progresivo en el *standard error* cada <n> registros de ligas cargados.

Archivos de ligas de longitud fija

+fix[/m]

Cargar archivos de longitud fija

+fix

Carga archivos de ligas con registros de longitud fija con el formato:

```
KEY MFN TAG OCC CNT
```

Carga archivo de ligas con formato reducido

+fix/m

Carga archivos de ligas con registros de longitud fija con el formato:

```
KEY MFN
```

IFLOAD - Salida

El archivo invertido consiste de seis archivos físicos, cinco de los cuales contienen el diccionario de términos recuperables (organizados como un árbol B*) y el sexto contiene la lista de postings asociados con cada término. A los efectos de optimizar el almacenamiento en disco, se mantienen dos árboles B* (estructura de datos que permite almacenar información clasificada) separados, uno para términos de hasta 10 caracteres y otro para términos de más de 10 caracteres y hasta un máximo de

30 caracteres. Ambos árboles B* están estructurados como páginas de longitud fija, con claves completadas con espacios en blanco a la derecha.

Puede obtenerse una optimización de espacio de disco adicional usando el Programa Utilitario de CISIS MYZ, que comprime el diccionario del archivo invertido para cada uno de los árboles B*, como se indica a continuación:

```
myz ifn 1 ifn_z tell=10
myz ifn 2 ifn_z tell=10
```

El diccionario del archivo invertido resultante ifn_z está conformado por los archivos *ifn_z.cnt, ifn_z.n01, ifn_z.l01, ifn_z.n02 y ifn_z.l02*.

Después de que se ejecutan los comandos el archivo ifn.ifp deberá renombrarse como *ifn_z.ifp*, donde ifn_z es la versión comprimida del archivo invertido inf.

IFUPD - Programa

El programa IFUPD actualiza un archivo invertido, de acuerdo a lo establecido en:

- La Tabla de Selección de Campos (FST);
- Stopwords;
- Los registros del archivo maestro;
- Las opciones de procesamiento.

El programa IFUPD puede también usarse para crear y reinicializar un archivo invertido.

IFUPD - Presentación

El comando:

```
ifupd \cisis\bases\cds fst=@ stw=@
```

Lee el archivo maestro cds localizado en el directorio \cisis\bases y actualiza el archivo invertido correspondiente, de acuerdo a la Tabla de Selección de Campos por defecto (FST) y a los archivos *Stopword -cds.fst* y *cds.stw*- localizados en el mismo directorio.

El comando siguiente usa una Tabla de Selección de Campos diferente y especifica que las claves del archivo invertido a ser extraídas no usan *Stopwords*:

```
ifupd \cisis\bases\cds fst=@\cisis\bases\another.fst
```

IFUPD acepta especificaciones en la línea de entrada, tal como:

```
ifupd \cisis\bases\cds "fst=70 0 (v70/); 69 2 v69"
```

IFUPD permite que sean actualizados varios archivos invertidos para un archivo maestro dado. En este caso, todas las actualizaciones (excepto la última) deben especificar que la señal de *I/F update is pending* debe mantenerse, como se indica a continuación:

```
ifupd au fst=@au.fst master=\cisis\bases\cds reset=0
ifupd ti fst=@ti.fst master=\cisis\bases\cds reset=0
ifupd kw fst=@kw.fst master=\cisis\bases\cds
```

En la última línea del ejemplo no aparece reset=0 debido a que se actualizaron todos los archivos invertidos, por lo tanto no es necesario mantener la señal de actualización pendiente.

IFUPD - Sintaxis

```
ifupd [mono|full] [create=]<ifname> [<option> [...]]

[mono|full]                          → single/multi user operation
[create=]                            → to delete+create <ifname>
<ifname>                             → output inverted file

options:

 fst=<fstspec>|@[fstfile]            → field select table
 stw=<stwspec>|@[stwfile]            → stop words
 -posts                              → (init and) do not load .ifp
 master=<name>                       → alternate master file
 actab=<file>                        → alphabetic chars table
 uctab=<file>                        → upper case chars table
 from=<n>                            → initial mfn
```

```
to=<n>          → final mfn
count=<n>       → max mfns
tell=<n>        → tell <n>% loaded
```

Parámetros obligatorios

<ifname>

Archivo invertido a ser actualizado

<dbname>

Archivo invertido a ser actualizado.

Si se especifica *create=<dbname>*, se crea el archivo invertido <*dbname*> y se reinicializa incluso si éste ya existe.

Tabla de selección de campos

fst=<fstspec>|@[fstfile]

FST por defecto

fst=@

Usa la Tabla de Selección de Campos (FST) por defecto.

Cargar FST desde un archivo externo

fst=@<file>

La Tabla de Selección de Campos (FST) se suministra en el archivo <file>.

Especificación de FST en línea

fst=<fstspec>

La Tabla de Selección de Campos (FST) se suministra en la línea de entrada como una lista separada por punto y comas.

Archivo de palabras no significativas

stw=<stwspec>|@[stwfile]

Archivo STW por defecto

stw=@

Usa el archivo *Stopword* por defecto.

Cargar lista STW desde un archivo externo

stw=@<stwfile>

La lista de palabras no significativas (*stopwords*) se suministra en el archivo <stwfile>.

Mantener marca de actualización pendiente

reset=0

Mantiene la señal de *I/F update is pending*; por defecto reinicializa todos los registros procesados.

No cargar postings

-posts

Actualiza sólo el diccionario. No carga los postings en el archivo *.ifp*.

Archivo maestro alternativo

master=<name>

Por defecto asume como archivo maestro uno que tenga el mismo nombre que el archivo invertido que se está procesando, pero si se indica *master=<name>*, *ifupd* utilizará un archivo maestro llamado <name>.

IFUPD - Salida

Solamente se procesan los registros del archivo maestro que tienen la señal de *I/F update is pending*.

El parámetro *reset=0* permite que los registros procesados del archivo maestro puedan posteriormente usarse para crear o actualizar otro archivo invertido.

MYS - Sintaxis

```
mys {link1|link2|<preclen>} <fileln> <filelk> [<option> [...]]

options:  tell=<n>
          +fix/m

Ex: mys 37 x.ln1 x.lk1
     mys link1 x.ln1 x.lk1 tell=0

Ex: mys 57 x.ln2 x.lk2
     mys link2 x.ln2 x.lk2 tell=0
```

MYS – Salida

Hace un sort del archivo de ligas (links) para crear el archivo invertido

IFMERGE - Sintaxis

```
ifmerge <out> <if1>[,n1] <if2>[,n2] [...] [<option> [...]]
 <out>                    → output inverted file
 <if1>[,n1]               → input inverted file #1, maxmfn#1
 <if2>[,n2]               → input inverted file #2, maxmfn#2

options:
+mstxrf                   → create <out> M/F and its mstxrf <out>.pft
-posts                    → do not load .ifp
-balan                    → do not rebalance the dict
tell=<n>                  → tell <n> keys has been loaded
```

IFMERGE – Salida

Combina varios archivos invertidos de diferentes archivos master en un solo archivos invertidos, con un procedimiento para recuperar los registros desde los archivos master fuentes.

MKIY0 - Sintaxis

```
mkiy0 <dbn> [-ifp] [-v] [blksize=32768]
```

MKIY0 - Salida

Combina los seis archivos que componen el archivo invertido en un solo archivo físico.

CRUNCHIF - Sintaxis

```
crunchif <dbn> <target_dbn> [<option> [...]]

options:
```

```
-ifp                          → don't crunch .ifp file
/ifp                          → crunch .ifp file if needed
tell=<n>                      → tell <n> records processed

target={linux|hpux|sun|alpha|vax|unisys|mpe|cdc|pc}  default: linux
```

CRUNCHIF - Salida

Convierte el archivo invertido de un sistema operativo a otro, por ejemplo de Windows a Linux.

Referencias bibliográficas

1. UNESCO. *Mini-micro CDS/ISIS:* Reference manual (version 2.3). Organized by Giampaolo Del Bigio. Paris: United Nations Educational, Scientific and Cultural Organization, 1989. 286 p. ISBN 92-3-102-605-5.

2. BUXTON, Andrew, HOPKINSON, Alan. *The CDS/ISIS for Windows Handbook* [online]. Paris: United Nations Educational, Scientific and Cultural Organization, 2001 [cited 30 August 2006]. 164 p. Available from internet: <http://bvsmodelo.bvs.br/download/winisis/winisis-handbook-en.pdf>.

3. SUTER, Tito. "Prehistoria" e historia del MicroISIS [online]. In: *Manual para instructores de Winisis*. Buenos Aires: Centro Atómico Constituyentes (CAC), Comisión Nacional de Energía Atómica (CNEA), 1999 [citado el 30 Agosto 2006]. p. 21-26. Disponible en internet: <http://www.cnea.gov.ar/cac/ci/isis/isidams.htm>.

Glosario

- **Archivo**. En computación, un conjunto de datos que se puede grabar en algún dispositivo de almacenamiento. Los archivos de datos son creados por aplicaciones, como por ejemplo un procesador de textos.

- **Archivo invertido**. Conjunto de seis archivos físicos, cinco de los cuales contienen los términos de búsqueda del diccionario (organizados como un árbol B*) y el sexto contiene la lista de apuntadores asociadas a cada término. A fin de optimizar el almacenamiento en disco, se mantienen dos árboles B* por separado: uno para los términos de hasta 10 caracteres (almacenados en los archivos *.N01* y *.L01*) y otro para los términos de más de 10 caracteres (almacenados en los archivos *.N02* y *.L02*). El archivo *.CNT* contiene campos de control para ambos árboles B*. En cada archivo del árbol B* el archivo *.N0x* contiene los nodos del árbol y el archivo *.L0x* contiene las hojas. Los registros de las hojas apuntan al lugar donde se encuentran los apuntadores que contienen la información para localizar los registros (postings) en la base de datos. Este archivo se identifica con la extensión *.IFP*.

- **Backup**. Procedimiento en el que uno o más archivos y/o directorios son duplicados para otro dispositivo de almacenamiento (cinta o disco), para producir una copia de seguridad, que puede restaurarse en el caso de que algún dato sea borrado accidentalmente o si ocurrió daño físico de los datos originales.

- **Base de datos**. Colección de datos estructurados para que sea posible acceder a ellos y manipularlos fácilmente. Está formada por unidades denominadas registros, cuyos diversos atributos son representados por campos y subcampos. Por ejemplo, en un archivo "catastro de clientes", cada cliente representa un registro, que posee varios campos, como "NOMBRE", "CÓDIGO DEL CLIENTE", "TELÉFONO" etc.

- **Bases de datos bibliográfica**. Versión electrónica de un catálogo o índice bibliográfico.

- **Campo**. Elemento de un registro que permite almacenar información específica. Ver *Base de datos*.

- **CDS/ISIS – MicroISIS**. Software desarrollado y mantenido por la UNESCO para el tratamiento de datos bibliográficos.

- **Clave**. Expresión que identifica una o más informaciones de determinada clase o tipo y que puede ser usada en la busqueda.

- **Formato de presentación**. Conjunto de comandos que determinan como debe ser la salida de datos de una base de datos ISIS.

- **Formato electrónico**. Cualquier forma de almacenamiento, recuperación y presentación de información pasible de transmisión online o grabación en medios magnéticos u ópticos.

- **Formato ISO (de intercámbio de datos)**. Patrón establecido por la ISO para intercambio de datos entre instituciones, redes y usuarios. Se refiere la norma ISO 2709.

- **Formato LILACS**. Formato de descripción bibliográfica establecido por BIREME, basado en la UNISIST Reference Manual for Machine-readable Bibliographic Descriptions.

- **Indización**. Procedimiento de identificar y describir el contenido de un documento con términos que representan los temas correspondientes a ese documento, con el objetivo de recuperarlo posteriormente.

- **Posting**. Consiste de la dirección de una clave extraída del archivo maestro.

- **Registro**. Conjunto estructurado de datos que permite almacenar determinado asunto. Ver *Base de datos*.

- **Subcampo**. Elemento que contiene la menor parte de información de un campo, cuyo sentido puede no ser claro si no fuera analizado en conjunto con los otros elementos relacionados.

- **UNISIST**. Programa intergubernamental relativo a las cooperaciones en el campo de la información científica y tecnológica.

Apéndice I - Parámetros de uso general

Relativos a la salida estándar:

now, tell, -all, >, >>

Deshabilitar *prompt* entre registros

now (nowait)

Este parámetro deshabilita el prompt que presenta MX entre registro y registro.

El parámetro *now* o *nowait* permite que se procesen todos los registros sin intervención del operador y suprime la salida del indicador *(prompt)* del programa.

Ahora se obtendrá la salida completa en forma inmediata sin que se detenga entre registro y registro con el indicador de dos puntos .. (*prompt)*:

```
mx cds from=24 to=50 pft=mfn/ now
```

Informar cada n registros

tell=<n>

El parámetro *tell=n* produce un mensaje breve en el dispositivo de salida de mensajes de error (*stderr*, normalmente la pantalla) cada <n> registros.

Aunque el proceso se redireccione a una salida impresa o a un archivo, los mensajes producidos por *tell=n* no pueden ser redireccionados y seguirán saliendo por la pantalla.

Se recomienda usar este parámetro en conjunto con el parámetro *-all* que suprime de la salida en pantalla el listado de los registros.

Ejemplo:

```
mx cds from=1 to=150 tell=30 now -all
```

Producirá en la pantalla de la computadora sólo los siguientes mensajes:

```
C:\>mx cds from=1 to=150 tell=30 now -all
+++ 30
+++ 60
+++ 90
+++ 120
+++ 150
C:\>
```

Deshabilitar volcado de información en pantalla

-all

El parámetro *-all* ocasiona que no se envíe a la pantalla ningún tipo de información, salvo mensajes de error o información que genere el parámetro *tell* (que utiliza el *standard error* para enviar sus mensajes).

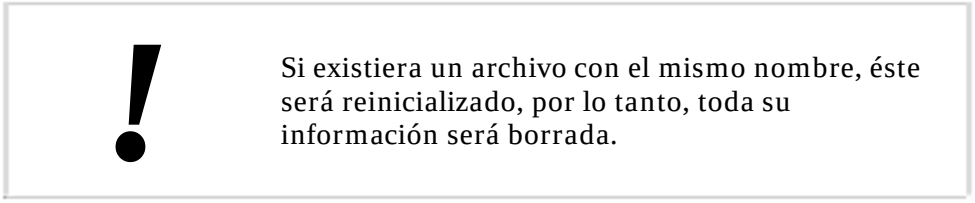
Redireccionar salida estándar

> <file> | >> <file>

Toda la información presentada en pantalla (salida estándar por defecto, *standard output*), puede ser redirigida a un archivo, impresora, etc.

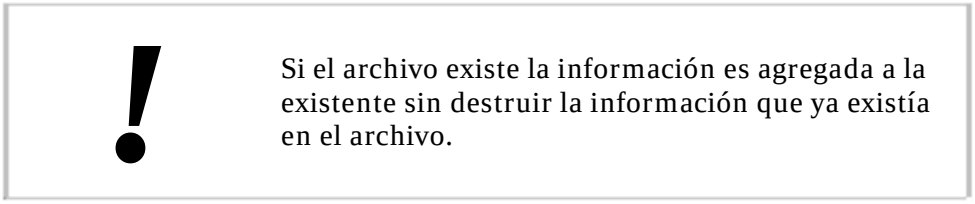
Es posible dirigir la salida a un archivo o a una impresora usando los atributos de redireccionamiento del sistema operativo: > o >>.

Si se agrega al final de la línea de comandos la instrucción >***file***, MX crea un archivo con nombre ***file*** y guarda en él toda la información que estaba dirigida a la salida estándar (pantalla).



Si existiera un archivo con el mismo nombre, éste será reinicializado, por lo tanto, toda su información será borrada.

Si se coloca >>***file*** en la llamada del MX, éste abre el archivo con nombre ***file*** y agrega toda la información que iba dirigida a la salida estándar (pantalla), si el archivo no existe es creado.



Si el archivo existe la información es agregada a la existente sin destruir la información que ya existía en el archivo.

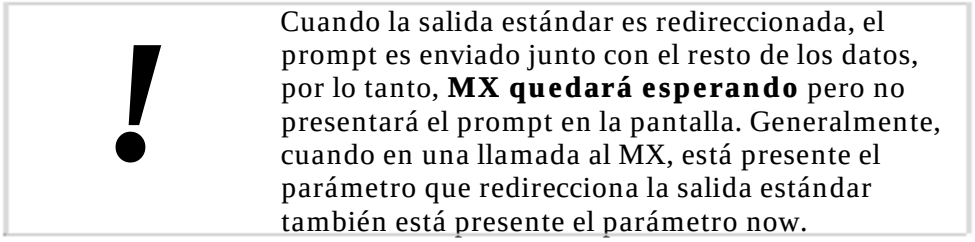
Ejemplos:

- Enviar el listado del resultado de una búsqueda a la impresora:

```
mx cds plants pft=@cds.pft now > LPT1
```

- Enviar el listado de un rango de registros a un archivo:

```
mx cds from=10 to=30 pft=@otro.pft lw=35 now > archivo.txt
```



Cuando la salida estándar es redireccionada, el prompt es enviado junto con el resto de los datos, por lo tanto, **MX quedará esperando** pero no presentará el prompt en la pantalla. Generalmente, cuando en una llamada al MX, está presente el parámetro que redirecciona la salida estándar también está presente el parámetro now.

Tome en cuenta que si se coloca el parámetro ***-all*** no habrá salida, entonces el archivo al que se direccionó la salida quedará vacío.

Relativos a la selección de registros:

<from> <to> <count> <loop>

Iniciar en registro n

from=<n>

Comienza el proceso en el registro <n> de la base de datos de entrada; si no se especifica, el proceso comienza en el primer registro del archivo maestro.

Finalizar en registro n

to=<n>

Finaliza el proceso en el registro <n> de la base de entrada; si no se especifica, finaliza en el último registro del archivo maestro.

Procesar un registro cada n

loop=<n>

Saltea <n> registros por cada registro procesado.

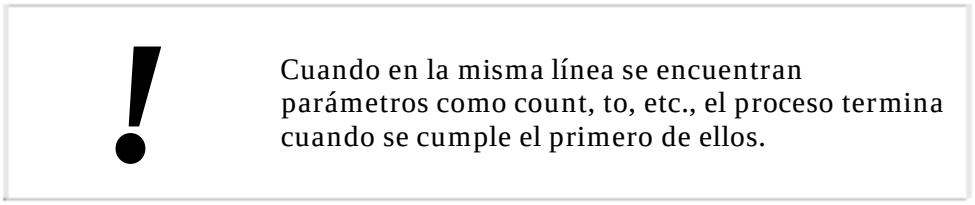
Por ejemplo, si este parámetro está presente y el valor precedido por el igual es 5, se procesarán los registros: 1, 6, 11, etc.

Seleccionar *n* registros

count=<n>

El parámetro *count=n* selecciona exactamente *n* registros a partir de un inicio dado. Si no se indica registro de inicio comienza desde el primer registro.

Ejemplo con MX	**Salida**
`mx cds from=24 to=50 loop=5 pft=mfn/ now`	000024 000029 000034 000039 000044 000049
`mx cds from=24 to=50 count=3 loop=5 pft=mfn/ now`	000024 000029 000034
`mx cds from=24 to=50 count=9 loop=5 pft=mfn/ now`	000024 000029 000034 000039 000044 000049



Cuando en la misma línea se encuentran parámetros como count, to, etc., el proceso termina cuando se cumple el primero de ellos.

Relativos a los registros de salida:

<offset>

Sumar n a los números de registro

offset=<n>

Agrega <n> al MFN; así el MFN que se guarda en *dbout* es el *MFN* del registro de entrada más *offset*.

```
mxcp cds newcds offset=1000
```

En el ejemplo, al indicarle offset=1000, se ingresarán los registros de la base cds en la base newcds con mfn 1000, 1001, 1002, etc.

Cambio global de patrones

gizmo

El parámetro *gizmo*= permite realizar cambios globales en el contenido de los campos de una base CDS/ISIS, convertir una cadena de caracteres en otra, y así realizar modificaciones, codificación/decodificación, compresión de datos, etc.

Estos cambios pueden realizarse sobre todos los registros de la base o un conjunto de registros (seleccionados por medio de una búsqueda, un rango, etc.). A su vez, los cambios pueden abarcar a todo el registro o sólo a algunos campos. Por ejemplo, para cambiar los signos < > por / /, o la cadena de caracteres *plants* por *PLANTAS*, etc.

Para efectuar estos cambios es necesario disponer de un archivo maestro *gizmo*. Este archivo maestro tiene en principio dos campos: el campo 1 contiene el dato a cambiar, y el campo 2 el nuevo valor. Cada pareja de datos será un registro de la base *gizmo*.

Cada registro de entrada se somete al procedimiento de cambio establecido en el archivo *gizmo*. Al comenzar la ejecución de MX los datos del archivo gizmo se cargan como una tabla en memoria y se ordenan alfabéticamente por el valor del campo 1 y por su largo (de esta manera las cadenas de caracteres más largas son convertidas antes que las cortas).

Ejemplos:

Se crea una base de datos llamada PRUEBA usando los parámetros de MX conocidos y se ingresan los datos directamente desde el teclado:

```
mx seq=con create=prueba -all now
<|/
>|/
plants|PLANTAS
<ctrl>Z (o <F6>)
```

Obteniendo los registros siguientes:

```
mfn= 1
1 «<»
2 «/»
mfn= 2
1 «>»
2 «/»
mfn= 3
1 «plants»
2 «PLANTAS»
```

El contenido de los campos título y descriptores del registro MFN=1 es:

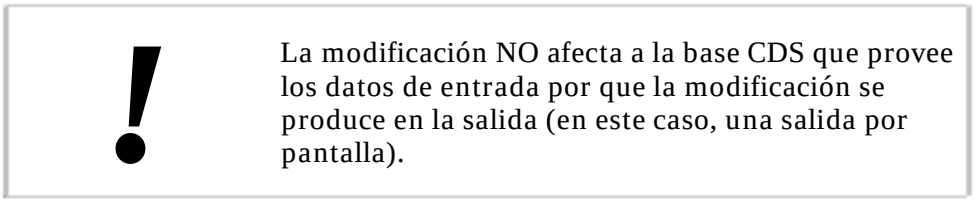
```
mx cds to=1 "pft=mfn/v24/v69"
000001
Techniques for the measurement of transpiration of individual plants
Paper on: <plant physiology><plant transpiration><measurement and instruments>
```

Si al siguiente ejemplo se aplica el parámetro gizmo:

```
mx cds gizmo=prueba to=1 "pft=mfn/v24/v69"
```

Dará como resultado:

```
000001
Techniques for the measurement of transpiration of individual PLANTAS
Paper on: /plant physiology//plant transpiration//measurement and instruments/
```



Para modificar realmente los registros se debería enviar el resultado del proceso al mismo archivo maestro, como en el ejemplo siguiente:

```
mx cds gizmo=prueba to=1 copy=cds –all now
```

Si en cambio se desea generar una nueva base de datos es necesario crearla:

```
mx cds gizmo=prueba to=1 create=salida –all now
```

La modificación es posible restringirla a un campo específico del registro, indicando a continuación del parámetro *gizmo* la etiqueta o etiquetas sobre las que se producirá el cambio. Es posible indicar un rango de etiquetas separadas por /.

```
mx cds gizmo=prueba,69,24 to=1 create=salida -all now
mx cds gizmo=prueba,35/56 to=1 create=salida -all now
```

Apéndice II - Archivo CIPAR

La interface de programación CISIS provee una herramienta llamada CIPAR, que hace las veces del SYSPAR.PAR y de los <dbn>.par del CDS/ISIS estándar, además de muchas otras funciones propias del CISIS. El CIPAR es un archivo ASCII puro que activa mecanismos para la búsqueda de nombres de archivos lógicos, nombres de bases de datos, establece parámetros del entorno de trabajo, etc.

Un archivo de parámetros CIPAR consiste de líneas de comandos, un comando por línea, con instrucciones de equivalencias lógicas. Una equivalencia lógica es una sentencia de asignación donde el valor a la izquierda del signo = representa al conjunto de valores a la derecha del signo.

CISIS lee las líneas del archivo CIPAR y las almacena en memoria como una tabla de referencias. Cada línea es leída hasta que se llega al fin de archivo o hasta que se encuentra la combinación de /* (barra, asterisco), lo que ocurra primero. El texto que sigue luego del /* se considera como comentario y documentación sobre el CIPAR.

Ejemplos:

Suponiendo que el archivo se llama DATOS.CIP y contiene las siguientes líneas:

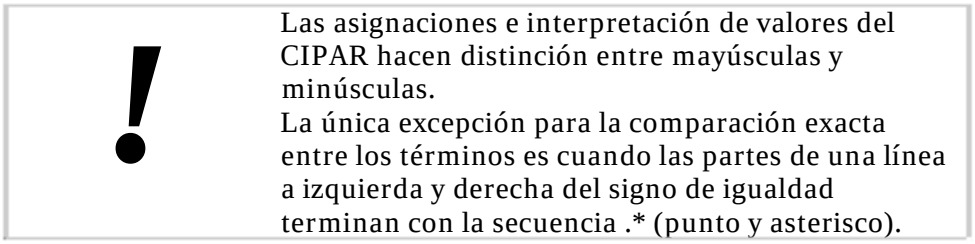
```
CDS.*=\cisis\bases\cds.*
CDS1.PFT=\cisis\bases\cds1.pft
```

Entonces la instrucción

```
mx cipar=datos.cip CDS pft=@CDS1.PFT from=10 ...
```

Es equivalente a:

```
mx \cisis\bases\cds pft=\cisis\bases\cds1.pft from=10 ...
```



En el ejemplo siguiente:

```
dbxtrace=y
14=1
cds.*=c:\cisis\bases\cds.*
cds1.pft=c:\cisis\bases\cds1.pft
lilacs.xrf=c:\lilacsok.xrf
lilacs.*=d:\bases\lilacs\lilacs.*
```

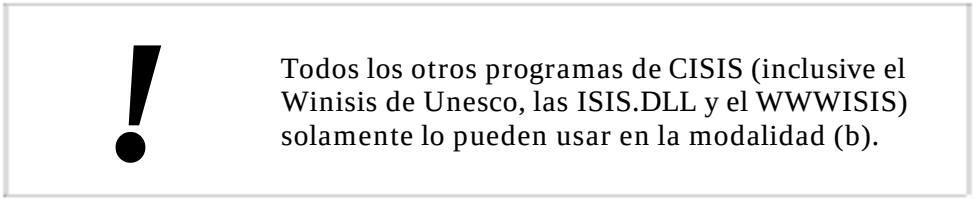
Los términos a la izquierda serán o no convertidos en los términos de la derecha según se cumpla o no con la comparación exacta. Si el programa MX encuentra una secuencia de caracteres (que pudiera ser el nombre de una base de datos, de un formato, etc.), buscará en el las líneas del CIPAR y realizará las transformaciones que correspondieren si encuentra la línea de equivalencia.

Encuentra	Convierte a:
dbxTRACE	No lo convierte
dbxtrace	y
14	1
cds.mst	c:\cisis\bases\cds.mst
lilacs.xxx	d:\bases\lilacs\lilacs.xxx
lilacs.xrf	c:\lilacsok.xrf
LILACS.xxx	No lo convierte

Además de la definición de nombres lógicos de archivos es posible asignar valores de variables de ambiente (*environment*) en forma global a las aplicaciones.

Hay dos maneras de activar el CIPAR:

a) Usándolo directamente dentro de la línea de comandos de MX mediante el parámetro *cipar=<file>*, donde *<file>* especifica el nombre de archivo que se usará como CIPAR durante la ejecución del programa. Sólo el programa MX puede usar esta opción.



Todos los otros programas de CISIS (inclusive el Winisis de Unesco, las ISIS.DLL y el WWWISIS) solamente lo pueden usar en la modalidad (b).

b) Como variable de ambiente del sistema operativo. Para esto se usará el comando *set=CIPAR=<nombre>* donde *<nombre>* será el archivo CIPAR que se usará en todas las ejecuciones posteriores de los programas CISIS.

Ejemplo:

```
c:\>set cipar=\cipar.par
```

Esta instrucción creará una variable de ambiente (una variable global del sistema operativo) llamada CIPAR cuyo contenido es c:\cipar.par. Todas las aplicaciones CISIS que se ejecuten a partir de este momento tomarán a ese archivo como CIPAR, no importa desde dónde se corran los programas.

Es posible cambiar el parámetro a otro archivo reasignando la variable de ambiente nuevamente con una instrucción set=<nuevo archivo>.

```
c:\> set cipar=\otro.par
```

Para eliminar esa variable del ambiente del sistema operativo es suficiente con asignarle un valor nulo:

```
c:\> set cipar=
```

La capacidad de asignar valores globales del sistema operativo no es exclusivo del MS-DOS; otros sistemas operativos tales como Unix también disponen de esta posibilidad, aunque usan otros comandos:

- SunOs y UNIX/CSH usan
  `setenv cipar=x`
- HP-UX, UNISYS 6000 y UNIX/ksh usan
  `cipar=x; export cipar`

Si existe CIPAR como variable de ambiente y además se declara el parámetro *cipar=* en la línea de comandos del MX, este último prevalecerá para la ejecución específica.

Ejemplo:

```
C:\> set cipar=c:\cipar.par
C:\> mx cipar=otro.par CDS from=10 to= ... etc.
```

La ejecución del MX de ese comando usará las definiciones del archivo **otro.par**.

Parámetros que se pueden incluir en el CIPAR

Parámetros sólo para MX

[cgitag=<tag>] [cgipfx=<pfx>] cgi={<fmt>|mx}

Recibe parámetros de una llamada a un CGI

Almacena los datos del CGI en un campo repetible de etiqueta definida por *cgitag*, por defecto el campo 2000. Cada pareja de datos nombre/valor se guarda en los subcampos ^n ^v. MX leerá cada línea mediante un formato (<fmt> o mx.pft) que deberá incluirse como parámetro adicional.

cgitag=<tag>	Etiqueta de campo para los subcampo n, v (por defecto 2000)
cgipfx=<pfx>	Prefijo del nombre de la etiqueta de campo (por defecto **tag**)
cgi=<fmt>	Especifica los parámetros del mx mediante un formato
cgi=mx	Especifica los parámetros del mx por los nombres correspondientes

Ejemplos:

- En la línea de comando se definen las variables REQUEST_METHOD=GET y QUERY_STRING, entonces se llama al mx indicando el formato a ser interpretado en modo CGI.
  - Para Windows

    ```
    set REQUEST_METHOD=GET
    set "QUERY_STRING=db~cds&count~2&now&btell~0&bool~plants*water&pft~mfn/"
    ```

  - Para Linux

    ```
    export REQUEST_METHOD=GET
    export QUERY_STRING="db~cds&count~2&now&btell~0&bool~plants*water&pft~mfn/"
    ```

  - Para Linux y Windows:

    ```
    mx cgi=mx
    000004
    000011
    ```

Para que el formato interprete la variable *QUERY_STRING*, se debe utilizar como separador de ocurrencias el carácter reservado & y como separador de subcampos el carácter ~. El formato especificado debe tratar los subcampos *n* (anterior al ~ en caso de que exista) y *v* (posterior al ~ en caso de que exista) de la etiqueta definida de campo (tag).

- En la linea de comando se definen las variables REQUEST_METHOD=GET, QUERY_STRING, entonces se llama al mx forzando la lectura de datos en un campo determinado (11000 en el ejemplo) e indicando el formato a ser interpretado por el modo CGI.
  - Para Windows

```
set REQUEST_METHOD=GET
set "QUERY_STRING=db~cds&count~2&now&btell~0&bool~plants*water&pft~mfn/"
```

  - Para Linux

```
export REQUEST_METHOD=GET
export QUERY_STRING="db~cds&count~2&now&btell~0&bool~plants*water&pft~mfn/"
```

  - Para Linux y Windows:

```
mx cgitag=11000   “cgi=(if v11000^n=’db’ then v11000^n,’=’,v11000^v/ fi)”
mfn=1
 24 «Techniques for the measurement of transpiration of individual plants»
 26  «^aParis^bUnesco^c-1965»
 30  «^ap. 211-224^billus.»
 70  «Magalhaes, A.C»
 70  «Franco, C.M.»
 44  «Methodology of plant eco-physiology: proceedings of the Montpellier
Symposium»
 ...
```

- En este ejemplo los parámetros son pasados por el ambiente CGI y no por las variables de ambiente. En un servidor web, con el programa mx y el formato mx.pft grabados en el área de CGI, se monta la página siguiente.

```
<html>
   <head>
      <title>Ejemplo de llamada de MX en ambiente CGI</title>
   </head>
<body>
   <form method="post"  action="/cgi-bin/mx.exe/cgi=mx" >
      Ejecutar MX con  'bool=' y 'count='  atribuyendo los siguientes valores:
<hr>
      Expresión de búsqueda (bool=): <input type="text" name="bool"
value="$"></input> <br/ >
```

```
      Num. de registros a exhibir (count=): <input type="text" name="count"
value="10">
         </input> <br/ >
      <input type="hidden" name="db" value="d:\httpd\bases\cds\cds”></input>
      <input type="hidden" name="now"></input>
      <input type="hidden" name="btell" value="0"></input>
      <input type="hidden" name="pft"

value="lw(8000),newline('<br>'),@d:\httpd\bases\cds\cds”.pft"></input>
      <input type="submit" value="search & display"></input>
   </form>
</body>
</html>
```

En el ejemplo se ha usado el formato padrón *mx.pft*, como puede observarse en la sentencia *action* de la etiqueta <form>

cipar: ci_tempdir=<path>

Define el drive y/o directorio donde se crearán los archivos temporarios de trabajo. Si no se define el parámetro ci_tempdir, entonces los archivos temporarios se crearán en el directorio indicado por la variable de ambiente TEMP o TMP.

```
echo ci_tempdir=c:\work >xcip
mx cipar=xcip cdromdb

@set ci_tempdir=c:\work
mx cdromdb
```

cipar: maxmfrl=<nbytes>

Especifica el tamaño del MFRmfrl (master file record length). Por defecto tendrá un largo de 32767 bytes que es el máximo MFRmfrl estándar.

```
echo maxmfrl=32767 >xcip
mx cipar=xcip bigrecsdb
```

cipar: ci_fststrip=<maxlen>

Elimina cualquier marca del tipo <text> o </text> hasta un máximo de <maxlen> caracteres de largo para los campos de datos al comienzo de una ejecución de FST (los campos de datos son rellenados con espacios en blanco).

```
echo ci_fststrip=21 >xcip
mx cds
mfn= 1
69 «Paper on: <plant physiology><plant transpiration><measurement and
instruments>»
mx cipar=xcip cds "fst=690 4 v69"
mfn= 1
69 «Paper on: <measurement and instruments> »
690 «PAPER^m1^o1^c1^l1»
690 «ON^m1^o1^c2^l1»
690 «MEASUREMENT^m1^o1^c3^l2»
690 «AND^m1^o1^c4^l1»
690 «INSTRUMENTS^m1^o1^c5^l2»
```

Parámetros para MX y aplicaciones programadas para ambiente multiusuario

{netws | 14} | <dbn>.net ={0 | single | MONONETS} |

{1 | full | FULLNETS} |{ 2 | master | MASTNETS }

Los valores del parámetro se corresponden con los valores 0, 1, 2, del parámetro 14 del SYSPAR.PAR. El valor por defecto es MONONETS. También es posible asignar el parámetro de red a una base de datos específica mediante <*dbn*>.*net*=<*n*>

Ejemplos:

```
14=1
netws=MASTNETS
cds.net=FULLNET
```

Parámetro maxmfrl

maxmfrl=<n>

Realiza la misma especificación que el parámetro *mfrl=<n>* de la línea de comandos del programa MX.

Parámetro mstxl en el CIPAR

mstxl=<n>

mstxl | <dbn>/mstxl=<n>

n=0 ... 4 (0 es valor por defecto, predefinido en la Interfaz CISIS)

Determina el tamaño máximo que puede tener el archivo .MST. Este parámetro es leído y considerado solamente durante la creación de la base de datos.

<n>	Máximo .MST	Registro en bytes múltiplo de
0	512 MB	2 (valor por defecto)
1	1 GB	2
2	2 GB	4
3 o 4	4 GB	8

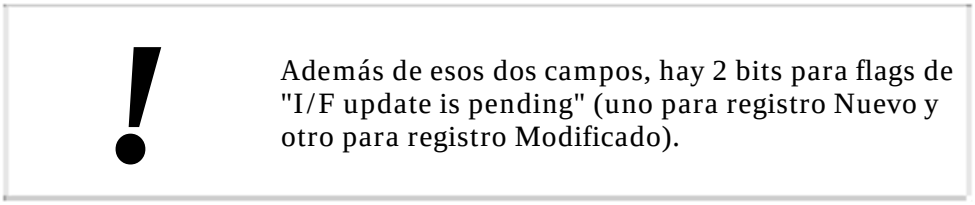
Ejemplo:

```
cds/mstxl=3
```

Estos parámetros también se aplican al WinISIS, ISIS.DLL y WWWISIS y a todas las aplicaciones desarrolladas con la Interfaz CISIS.

Cómo superar el límite de 512 MB para el archivo maestro:

El puntero de cada registro del .MST está almacenado en el .XRF. Este puntero se divide en dos campos: el campo número de bloque y el campo desplazamiento (*offset*) dentro del bloque. Estos dos campos ocupan respectivamente 20 y 9 bits.



Además de esos dos campos, hay 2 bits para flags de "I/F update is pending" (uno para registro Nuevo y otro para registro Modificado).

El puntero tiene 4 bytes y es tratado como entero con signo (un valor negativo indica físicamente/lógicamente borrado).

De esta forma el límite es de 512 Mb, como se indica más abajo:

```
2**20-1 = 1048675 bloques de 512 (offset desde 0 hasta 2**9-1 = 511)
```

La implementación del MSTXL en la Interfaz CISIS usa *offset* para bloques de 512 bytes en unidades de 2, 4 y 8 bytes (usando por lo tanto 8, 7 y 6 bits para almacenar el offset) y número de bloque en 21, 22, y 24 bits, respectivamente, para mstxl = 1, 2, y 3 (o 4).

La indicación de un archivo maestro en formato MSTXL está almacenada en el byte más a la izquierda del campo MFTYPE del registro de control del .MST, que se reinicializa cuando se lo carga en memoria a fines de procesamiento. Esta indicación se "recuerda" por CISIS durante las lecturas/grabaciones posteriores hasta que ese archivo maestro sea cerrado.

Es suficiente crear un archivo maestro con un CIPAR que contenga *mstxl*=<*n*> o <*dbn*>/*mstxl*=<*n*> para que los procesos subsiguientes respeten esa inicialización independientemente del CIPAR.

Parámetro dbxtrace=y

dbxtrace=y

Despliega mensajes en la salida estándar (*stdout - standard output*):

```
dbxopen - <dbn><.ext> fd=<n> [R]
dbxopen - <dbn><.ext> fd=<n> [RW]
```

<n>	file descriptor number
[R]	file was opened for read
[RW]	file was opened for read/write

dbxtrace=y idem "trace=dbx" del MX (como trace=rec, trm, giz, b40, par, mul, etc)

Parámetro mstload=<n>

mstload=<n>

Carga en memoria y cierra los archivos maestros (defecto=0).

```
mstload=<n>    idem "load=<n>" del MX, para M/F
```

Parámetro invload=<n>

invload=<n>

Carga en memoria y cierra los archivos invertidos (defecto=0).

```
invload=<n>    idem "load=<n>" del MX, para I/F
```

Parámetro mclose={y|n}

mclose={y|n}

Cierra todos los archivos maestros (defecto=n).

```
mclose=y     sólo permanece abierto un archivo maestro
```

Parámetro iflush={y|n}

iflush={y|n}

Vacía todos los archivos invertidos (defecto=n).

```
iflush=y     sólo permanece abierto un archivo invertido con su "data base
             descriptor" en memoria
```

Parámetro mflush={y|n}

mflush={y|n}

Vacía todos los archivos maestros (defecto=n).

`mflush=y`	sólo permanece abierto un archivo maestro con su "*data base descriptor*" en memoria. (*data base descriptor* incluye buffers de e/s y gizmos)

Parámetro what={y|n}

what={y|n}

Despliega la versión CISIS (defecto=n).

`what=y`	idem "what" del MX (llamado como único parámetro del MX)

Ejemplos

- Crear un archivo maestro sin registros o reinicializar una existente

```
mx seq=NUL create=NUEVA
```

Si existe una base de datos con ese nombre, MX no avisará que ésta será eliminada irreversiblemente.

- Compactar una base de datos

Este procedimiento recuperará el espacio perdido en el MST por las sucesivas actualizaciones de registros.

```
mx DATOS -all now create=DBN_AUX tell=100
xcopy DBN_AUX.* DATOS.*
del DBN_AUX.*
FULLINV DATOS DATOS.FST DATOS
```

- Verificar elementos duplicados

Se desea verificar en el catálogo de libros que no existan números de inventario duplicados. Los inventarios se registran en el campo 7 y se indizan con técnica 0 con el preliteral INV=, esto es: |INV=|v7.

```
mx LIBROS "pft=(if npost(|INV=|v7) > 1 then mfn,x3,v7/ fi)" tell=100 now >
duplic.lst
```

El procedimiento requiere que esté presente el archivo invertido.

- Hacer un análisis rápido de los campos usados en una base de datos

Son necesarios los programas MX.EXE y MXF0.EXE y estos dos archivos de formatos:

```
DMXF0A.PFT
'Analisis de los datos de la base ',v1001/#
" 1001 = input master file name ................. "v1001/,
" 1003 = date & time stamp ..................... "v1003/,
" 1009 = total number of records ............... "v1009/,
" 1010 = number of active records .............. "v1010/,
" 1011 = number of logically deleted records .... "v1011/,
" 1012 = number of phisically deleted records ... "v1012/,
DMXF0B.PFT
/#
"TAG DOCS OCCS"d1020/
"--------------------"d1020/
,(v1020^t,v1020^d,v1020^o/),
Con esto se prepara un archivo .bat, como el siguiente:
MYSCAN.BAT
REM %1 = <dbn_name> %2 = <nro regs estimados>
REM
mxf0 %1 create=lista %2 tell=100
mx lista pft=@dmxf0a.pft now > %1.lst
mx lista pft=@dmxf0b.pft now >> %1.lst
```

La ejecución de MYSCAN.BAT requiere dos parámetros de entrada, la base de datos con su path y la cantidad estimada de registros que contiene.

Como resultado se generará un archivo con el nombre de la base y extensión .lst.

Ejemplo:

```
myscan c:\dbisis\cds\cds 150
```

- Eliminar términos duplicados en un campo repetible

Suponiendo que los descriptores se registran en el campo v87, como campo repetible.

```
mx DATOS fmtl=20000 proc=@LIMPIO from=1 to=100 now -all create=OUT
```

El archivo LIMPIO tiene la siguiente especificación de formato:

```
proc('d870d871'),
( if v870[1] : s(v87|~|)
then
else proc('D870A870|'v870[1],v87'~|','A871|'v87'|')
fi ),
proc('d870'),
proc('d87d871',|A87~|v871|~|),
```

- Control de calidad de los datos

Se ofrece un modelo simplificado de un archivo .bat para realizar varios controles sobre una base de datos. Para este ejemplo se dispone de una base de datos llamada TEST a la que se le realizan los controles indicados en el menú, y una base THES (tesauro) contra la que se validan los descriptores de TEST.

```
CHK.BAT
@echo off
:BEGIN
cls
echo ------------------------------------------------------------
echo QUALITY CONTROL
echo ------------------------------------------------------------
echo.
echo E - Delete invalid characters (clean records)
echo O - Check mandatory fields
echo D - Check Descriptors
echo X - Exit
echo.
choice /c:EODX /N Select one option:
if errorlevel 4 goto END
if errorlevel 3 goto DESCRIPTORES
if errorlevel 2 goto OBLIGATORIOS
if errorlevel 1 goto CLEAN
:CONTINUA
echo.
echo (Press any key to continue...)
pause > nul
goto BEGIN
:CLEAN
```

```
call OPC_CLN. BAT
goto CONTINUA
:OBLIGATORIOS
call OPC_OBL.BAT
goto CONTINUA
:DESCRIPTORES
call OPC_DES.BAT
goto CONTINUA
:END
```

El procedimiento se completa con una serie de archivos .bat que realizan cada una de las opciones presentadas.

Los archivos se denominan OPC_xxx, donde xxx representa a cada una de las opciones del menú. Cada OPC_xxx activa un programa CISIS que llama a un archivo in=<text_file> que contiene los parámetros necesarios para la función de validación. Asociado a ese <text_file> hay un archivo de formato con el mismo nombre y con extensión .pft, que se usará para la validación de los datos.

Si la validación se realiza contra un archivo invertido, el in=<text_file> llamará a un procedimiento de jchk.

```
OPC_CLN.BAT
@echo off
mxcp \dbisis\TEST \dbisis\TEST clean > garbage
OPC_OBL.BAT
@echo off
echo Wait..
mx in=chk00 >> errores.tmp
CHK00
\dbisis\test
pft=@chk00.pft
-all
now
CHK00.PFT (ejemplo de un .PFT de validación)
if a(v02) then mfn,c8,'02->FATAL ERROR: field #2 missing!' fi/
if a(v01) then mfn,c8,'01->ERROR: field #01 missing!' fi/
if a(v923) and p(v23) then mfn,c8,'923->ERROR: field #923 missing!' fi/
if v31<>'ENG' then mfn,c8,'31->ERROR: field #31 invalid!' fi/
```

```
OPC_DES.BAT
@echo off
echo Wait...
mx in=chk620 >> errores.tmp
 CHK620 (verifica el campo 620 contra un tesauro)
\dbisis\test
jchk=Thes=mhu,(v620/)
pft=@chk620.pft
-all
now
 CHK620.PFT
if p(v32001) then ( if a(v32001^m) then mfn,c8,|620->|v32001^k,c65,'*invalid'/
fi ) fi
```

- Indice especial de títulos

Al indizar el campo de título (series o monografías) con técnica 0 (campo completo), la ordenación alfabética de éstos en el diccionario no respeta las normas bibliotecológicas de no considerar las palabras tales como: A, An, El, Los, The, etc.

En CDS/ISIS es posible obtener la salida impresa en forma correcta si se encierran esas palabras entre <..>, pero esta solución no está disponible para el archivo invertido. El siguiente ejemplo resuelve el problema.

Se indizará el campo v24 (título), entre los registros 15 y 20, de tres maneras para comparar los resultados. Los ejemplos 1 y 2 muestran lo que se obtiene mediante el CDS/ISIS estándar, y el ejemplo 3 muestra la solución del problema.

Se usa la FST siguiente: 240 0 v24

```
mx CDS from=15 to=20 now -all "fst=240 0 v24" ifupd/create=cds
ifkeys cds +tags
1| 240|<A> METHOD OF DETERMINING EVAP
1| 240|<THE> HEAT RESISTANCE OF PLANT
1| 240|<THE> MEASUREMENT OF DROUGHT R
1| 240|<THE> ROLE OF DEW IN PINE SURV
1| 240|GAUGES FOR THE STUDY OF EVAPOT
1| 240|MEASUREMENT OF DROUGHT RESISTA
```

Todos los títulos que tienen <...> aparecen al comienzo del diccionario pues el carácter < tiene un valor ASCII menor a la letra "A".

Se usa la FST siguiente: 240 0 mhu,v24

```
mx CDS "fst=240 0 mhu,v24" ifupd/create=cds now -all from=15 to=20
```

El resultado se presenta en el diccionario siguiente:

```
Ifkeys CDS +tags
1| 240|A METHOD OF DETERMINING EVAPOT
1| 240|GAUGES FOR THE STUDY OF EVAPOT
1| 240|MEASUREMENT OF DROUGHT RESISTA
1| 240|THE HEAT RESISTANCE OF PLANTS,
1| 240|THE MEASUREMENT OF DROUGHT RES
1| 240|THE ROLE OF DEW IN PINE SURVIV
```

Debido a que se usó MHU, los <..> no aparecen en el diccionario. La alfabetización considera las palabras no significativas.

La solución se obtiene usando un *gizmo*= WORDS, cuyo contenido es:

```
mfn= 1
1 «<The> »
mfn= 2
1 «<El> »
mfn= 3
1 «<An> »
mfn= 4
1 «<A> »
mfn= 5
1 «<L' >»
mfn= 6
1 «<LA> »
mfn= 7
1 «<La> »
mfn= 8
1 «<L'>»
..
```

No hay campo 2 en el gizmo, porque la conversión transforma el dato del campo 1 en una salida nula. Además se deja un espacio en blanco al final de cada campo para que los campos de títulos, luego de aplicado el gizmo, no comiencen con un espacio en blanco.

```
mx CDS from=15 to=20 now -all "fst=240 0 v24" ifupd/create=cds gizmo=words,24
ifkeys CDS +tags
1| 240|GAUGES FOR THE STUDY OF EVAPOT
1| 240|HEAT RESISTANCE OF PLANTS, ITS
2| 240|MEASUREMENT OF DROUGHT RESISTA
1| 240|METHOD OF DETERMINING EVAPOTRA
1| 240|ROLE OF DEW IN PINE SURVIVAL I
```

Apéndice III - Estructura de los registros de una base ISIS

Un registro con estructura ISIS tiene dos características especiales que ofrecen una gran versatilidad para el manejo de información textual: campos repetibles y de longitud variables.

Puesto que los registros no tienen un largo predeterminado, ni los campos tienen un largo fijo ni una cantidad predeterminada de repeticiones, no es posible tener acceso directo a ninguna porción de datos dentro de la base.

El acceso al registro se hace de modo indirecto, mediante punteros en un archivo auxiliar con extensión .XRF, y dentro del registro se accede a los datos mediante punteros en un directorio.

El archivo .XRF contiene toda la información necesaria para encontrar el punto de comienzo del registro solicitado dentro del .MST.

Para todos los ejemplos que siguen se usará el registro MFN=1 de la base CDS, cuyo contenido completo es el siguiente:

\cisis\bases\cds	Base de datos
Nxtmfn nxtmfb nxtmfp t reccnt mfcxx1 mfcxx2 mfcxx3 RC 152 123 13 0 0 0 0 0 0	Registro de control(control)
Mfn= 1|comb= 1|comp= 64| N| ... | 1+000=00000c40	xref

Mfn= 1|mfrl= 370|mfbwb= 0|mfbwp= 0|base= 66|nvf= 8|status= 0| 0	leader
Mfn= 1|dir= 1|tag= 44|pos= 0|len= 77 Mfn= 1|dir= 2|tag= 50|pos= 77|len= 11 Mfn= 1|dir= 3|tag= 69|pos= 88|len= 78 Mfn= 1|dir= 4|tag= 24|pos= 166|len= 68 Mfn= 1|dir= 5|tag= 26|pos= 234|len= 22 Mfn= 1|dir= 6|tag= 30|pos= 256|len= 20 Mfn= 1|dir= 7|tag= 70|pos= 276|len= 15 Mfn= 1|dir= 8|tag= 70|pos= 291|len= 12	dir
Mfn= 1 44 «Methodology of plant eco-physiology: proceedings of the Montpellier Symposium» 50 «Incl. bibl.» 69 «Paper on: <plant physiology><plant transpiration><measurement and instruments>» 24 «Techniques for the measurement of transpiration of individual plants» 26 «^aParis^bUnesco^c-1965» 30 «^ap. 211-224^billus.» 70 «Magalhaes, A.C.» 70 «Franco, C.M.» ..	fields

El Registro de CONTROL

Una base ISIS tiene un registro especial al comienzo (MFN=0) al que CDS/ISIS no provee acceso. Este registro tiene una estructura diferente del resto de los registros del archivo maestro (MST).

La información de este registro se muestra con el parámetro +control.

La estructura es:

Estructura del registro de CONTROL	
Ctlmfn	Siempre 0. Este campo no aparece con el MX <dbn> +control.
Nxtmfn	MFN a asignarse en el próximo registro.

Estructura del registro de CONTROL	
Nxtmfb	Último bloque asignado en el .MST. Los bloques son de 512 bytes.
Nxtmfp	Primera posición libre dentro del último bloque asignado. Un registro puede comenzar en cualquier posición libre entre 0-498 y extenderse por uno o más bloques. Ningún registro puede comenzar entre el byte 500 y 510.
Mftype	Tipo de base de datos: 0 base del usuario, 1 base de mensajes del sistema.
Reccnt	Reservado.
Mfcxx1	Reservado.
Mfcxx2	Bloqueo de ingreso de datos, valores 0, 1...n, depende de cuántos registros se estén editando en un cierto momento.
Mfcxx3	Bloqueo de lectura exclusiva, valores 0/1.

El Registro de XREF

El archivo .XRF está organizado como una tabla de punteros al archivo maestro (.MST). El primer puntero corresponde al MFN=1, el segundo al MFN=2, etc.

Cada puntero consiste de dos campos (4 bytes) que en la tabla del ejemplo anterior indican que el MFN=1 comienza en el bloque 1 (*comb*) y dentro del bloque a partir del byte 65 (*comp*) y que no tiene pendiente la actualización del archivo invertido. El último valor es la referencia real del puntero expresada en valor hexadecimal.

Cada bloque del .XRF es un archivo de 512 bytes de longitud y contiene 127 punteros (dato importante para la reconstrucción del .MST que se explica en el utilitario CTLMFN).

El Registro del Archivo MST

Los registros del archivo maestro se almacenan consecutivamente, uno tras otro, ocupando cada registro exactamente *MFRL* bytes. Cada archivo se almacena como bloques físicos de 512 bytes. Un registro puede comenzar en cualquier punto entre la posición 0 y 498 y puede extenderse por uno o más bloques.

El registro MST tiene longitud variable y consiste de tres secciones:

- Una de longitud fija (*leader*);
- Un directorio;
- Los campos de datos de longitud variable.

Estructura del LEADER

El *leader* consiste de un bloque de longitud fija de 18 bytes.

Mfn	Número de registro
Mfrl	Largo total del registro, incluyendo las tres secciones: *leader*, *directorio* y *área de datos*. Siempre es un número par.
Mfbwb	Puntero a la versión anterior del registro, número del bloque dentro del MST. Inicialmente está en 0, y también luego de la actualización del archivo invertido.
Mfbwp	Puntero a la versión anterior del registro. Desplazamiento dentro del bloque.
Base	Posición donde el área de datos comienza dentro del registro. Este valor es la suma del largo del *leader* más el largo del directorio.
Nvf	Cantidad de campos en el registro, o sea la cantidad de entradas en el directorio del registro.
Status	Indicador de borrado (0 = registro activo; 1 = lógicamente borrado)

Estructura del DIRECTORIO

El *directorio* del registro es un índice a los contenidos del registro en el segmento de datos. Este índice está constituido con tantas entradas como número de campos (nvf), permitiendo el acceso a los datos. Cada entrada en el *directorio* está formada por tres partes:

Tag	Identificador de campo o etiqueta.
Pos	Dirección en bytes donde comienza el primer carácter en el área de datos correspondiente a esta ocurrencia del campo.
Len	Largo del campo en bytes.

Apéndice IV - Lista de archivos TABs disponibles

Para usarse tanto en modo de presentación como para inversión de las bases

ASCII CODE PAGE 437 (CP437)

ac437.tab	Caracteres válidos del conjunto ASCII CP437
ac437n.tab	Caracteres válidos del conjunto ASCII CP437 incl. 0-9
ac437XT.tab	Caracteres válidos del conjunto ASCII CP437 incl. 0-9 e &'()*+,-./:
ma437.tab	Conversión a mayúsculas (con acento) de los caracteres
mi437.tab	Conversión a minúsculas (con acento) de los caracteres
na437.tab	Elimina acentos manteniendo mayúsculas/minúsculas de los caracteres
uc437.tab	Conversión a mayúsculas (sin acentos) de los caracteres
lc437.tab	Conversión a minúsculas (sin acentos) de los caracteres

ASCII CODE PAGE 850 (CP850)

ac850.tab	Caracteres válidos del conjunto ASCII CP850
ac850n.tab	Caracteres válidos del conjunto ASCII CP850 incl. 0-9
ac850XT.tab	Caracteres válidos del conjunto ASCII CP850 incl. 0-9 e &'()*+,-./:
ma850.tab	Conversión a mayúsculas (con acento) de los caracteres

mi850.tab	Conversión a minúsculas (con acento) de los caracteres
na850.tab	Elimina acentos manteniendo mayúsculas/minúsculas de los caracteres
uc850.tab	Conversión a mayúsculas (sin acentos) de los caracteres
lc850.tab	Conversión a minúsculas (sin acentos) de los caracteres

ANSI (Windows)

acans.tab	Caracteres válidos del conjunto ANSI
acansn.tab	Caracteres válidos del conjunto ANSI incl. 0-9
acansXT.tab	Caracteres válidos del conjunto ANSI incl. 0-9 e &'()*+,-./:
maans.tab	Conversión a mayúsculas (con acento) de los caracteres
mians.tab	Conversión a minúsculas (con acento) de los caracteres
naans.tab	Elimina acentos manteniendo mayúsculas/minúsculas de los caracteres
ucans.tab	Conversión a mayúsculas (sin acentos) de los caracteres
lcans.tab	Conversión a minúsculas (sin acentos) de los caracteres

GIZMOs disponibles para conversión del contenido de bases de datos

Conversión del conjunto de caracteres

g437ans	Conversión de ASCII CODE PAGE 437 para ANSI
gans437	Conversión de ANSI para ASCII CODE PAGE 437
g850ans	Conversión de ASCII CODE PAGE 850 para ANSI
gans850	Conversión de ANSI para ASCII CODE PAGE 850
gutf8ans	Conversión de UFT-8 para ANSI
gansutf8	Conversión de ANSI para UFT-8

ASCII CODE PAGE 437 (CP437)

g437ma	Conversión a mayúsculas acentuadas
g437mi	Conversión a minúsculas acentuadas
g437na	Retira acentos manteniendo mayúsculas/minúsculas
g437uc	Conversión a mayúsculas sin acento (Upper case)
g437lc	Conversión a minúsculas sin acento (Lower case)

ASCII CODE PAGE 850 (CP850)

g850ma	Conversión a mayúsculas acentuadas
g850mi	Conversión a minúsculas acentuadas
g850na	Retira acentos manteniendo mayúsculas/minúsculas
g850uc	Conversión a mayúsculas sin acento (Upper case)
g850lc	Conversión a minúsculas sin acento (Lower case)

ANSI (Windows)

gansma	Conversión a mayúsculas acentuadas
gansmi	Conversión a minúsculas acentuadas
gansna	Retira acentos manteniendo mayúsculas/minúsculas
gansuc	Conversión a mayúsculas sin acento (Upper case)
ganslc	Conversión a minúsculas sin acento (Lower case)

Conversión auxiliar de caracteres de marcación

gentit	Conversión de caracteres por la entidad HTML correspondiente ("&'<>)
gchar	Conversión a la entidad HTML por los caracteres correspondientes ("&'<>)

Conversión auxiliar de y para entidades HTML

Ghtmlans	Convierte entidades HTML en caracteres ANSI
ganshtml	Convierte caracteres ANSI en entidades HTML
ghtml850	Convierte entidades HTML en caracteres ASCII CP850
g850html	Convierte caracteres ASCII CP850 en entidades HTML
ghtml437	Convierte entidades HTML en caracteres ASCII CP437
g437html	Convierte caracteres ASCII CP437 en entidades HTML

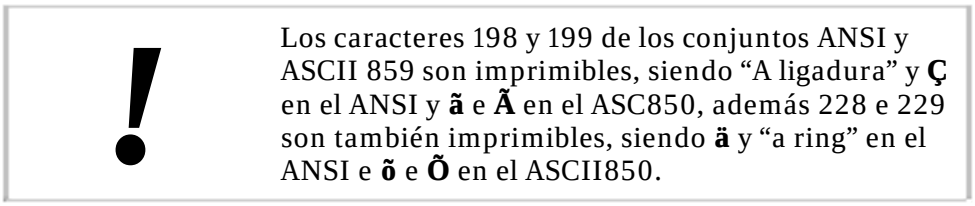
Cómo reconocer el conjunto de caracteres en que está una base CDS/ISIS

Si en la distribución de caracteres ofrecida por el `MXF0` hay AUSENCIA de caracteres con códigos entre 127(0x7f) y 159(0x9f) (inclusive), entonces pertenece al conjunto ANSI.

Si en la distribución de caracteres ofrecida por el `MXF0` hay una fuerte concentración entre los códigos 128(0x80) hasta 167(0xa7) (inclusive) es muy probable que sea Code Page 437.

Si en la distribución de caracteres ofrecida por el `MXF0` hay presencia de elementos en alguno de los siguientes códigos: 181(0xb5), 182, 183, 198, 199, 210, 211, 212, 214, 215, 216, 222, 224, 226, 227, 228, 229, 233, 234, 235, 236, 237 es muy probable que sea Code Page 859.

Es importante la presencia como caracteres discriminadores del código de página: el 198(0xc6) y 199(0xc6), que son **ã** y **Ã**, y asimismo 228(0xe4) y 229(0xe5) que son **õ** y **Õ**.



Los caracteres 198 y 199 de los conjuntos ANSI y ASCII 859 son imprimibles, siendo “A ligadura” y **Ç** en el ANSI y **ã** e **Ã** en el ASC850, además 228 e 229 son también imprimibles, siendo **ä** y “a ring” en el ANSI e **õ** e **Õ** en el ASCII850.

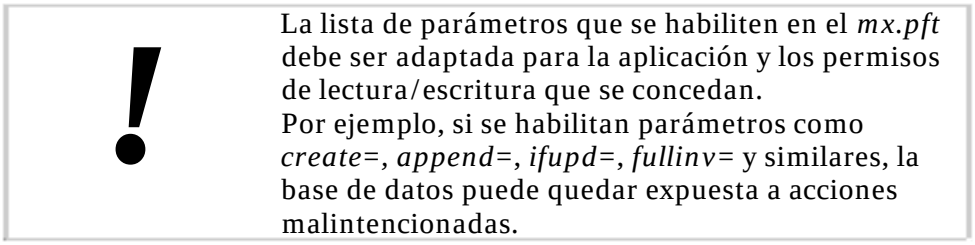
Elemento de complicación: Windows 95 y 98 utilizan codificación diferente de Windows 98SE y posteriores.

OBSERVACIONES

Los gizmos de conversión entre los conjuntos de caracteres: ANSI; ASCII CP437; y ASCII CP850, tiene como objetivo obtener caracteres gráficos similares.

Los gizmos de CONVERSION AUXILIAR se circunscriben solo a los caracteres acentuados.

Apéndice V - MX.PFT: Lista de parametros que extrae del environment de CGI



Parámetro	Formato de extracción
what	,(if v2000^n='what' then v2000^n/ fi),
prolog=	,(if v2000^n='prolog' then v2000^n'='v2000^v/ fi),
epilog=	,(if v2000^n='epilog' then v2000^n'='v2000^v/ fi),
in=	,(if v2000^n='in' then v2000^n'='v2000^v/ fi),
trace	,(if v2000^n='trace' then v2000^n/ fi),
trace=dbx	,(if v2000^n='trace=dbx' then v2000^n/ fi),
trace=rec	,(if v2000^n='trace=rec' then v2000^n/ fi),
trace=dec	,(if v2000^n='trace=dec' then v2000^n/ fi),
trace=trm	,(if v2000^n='trace=trm' then v2000^n/ fi),
trace=b50	,(if v2000^n='trace=b50' then v2000^n/ fi),
trace=b40	,(if v2000^n='trace=b40' then v2000^n/ fi),
trace=fmt	,(if v2000^n='trace=fmt' then v2000^n/ fi),
trace=fst	,(if v2000^n='trace=fst' then v2000^n/ fi),
trace=all	,(if v2000^n='trace=all' then v2000^n/ fi),

Parámetro	Formato de extracción
cipar=	,(if v2000^n='cipar' then v2000^n'='v2000^v/ fi),
mfrl=	,(if v2000^n='mfrl' then v2000^n'='v2000^v/ fi),
fmtl=	,(if v2000^n='fmtl' then v2000^n'='v2000^v/ fi),
load=	,(if v2000^n='load' then v2000^n'='v2000^v/ fi),
pages=	,(if v2000^n='pages' then v2000^n'='v2000^v/ fi),
uctab=	,(if v2000^n='uctab' then v2000^n'='v2000^v/ fi),
actab=	,(if v2000^n='actab' then v2000^n'='v2000^v/ fi),
-all	,(if v2000^n='-all' then v2000^n/ fi),
-control	,(if v2000^n='-control' then v2000^n/ fi),
-leader	,(if v2000^n='-leader' then v2000^n/ fi),
-xref	,(if v2000^n='-xref' then v2000^n/ fi),
-dir	,(if v2000^n='-dir' then v2000^n/ fi),
-fields	,(if v2000^n='-fields' then v2000^n/ fi),
+hits	,(if v2000^n='+hits' then v2000^n/ fi),
cgiplushits	,(if v2000^n='+hits''cgiplushits' then v2000^n'+hits'/ fi),
+fix	,(if v2000^n='+fix' then v2000^n/ fi),
+fix/m	,(if v2000^n='+fix/m' then v2000^n/ fi),
+all	,(if v2000^n='+all' then v2000^n/ fi),
+control	,(if v2000^n='+control' then v2000^n/ fi),
+leader	,(if v2000^n='+leader' then v2000^n/ fi),
+xref	,(if v2000^n='+xref' then v2000^n/ fi),
+dir	,(if v2000^n='+dir' then v2000^n/ fi),
+fields	,(if v2000^n='+fields' then v2000^n/ fi),
from=	,(if v2000^n='from' then v2000^n'='v2000^v/ fi),
to=	,(if v2000^n='to' then v2000^n'='v2000^v/ fi),
loop=	,(if v2000^n='loop' then v2000^n'='v2000^v/ fi),
count=	,(if v2000^n='count' then v2000^n'='v2000^v/ fi),
tell=	,(if v2000^n='tell' then v2000^n'='v2000^v/ fi),
b50=	,(if v2000^n='b50' then v2000^n'='v2000^v/ fi),
b40=	,(if v2000^n='b40' then v2000^n'='v2000^v/ fi),
nb1=	,(if v2000^n='nb1' then v2000^n'='v2000^v/ fi),
nbb=	,(if v2000^n='nbb' then v2000^n'='v2000^v/ fi),
nb0=	,(if v2000^n='nb0' then v2000^n'='v2000^v/ fi),
nb2=	,(if v2000^n='nb2' then v2000^n'='v2000^v/ fi),
btell=	,(if v2000^n='btell' then v2000^n'='v2000^v/ fi),
b50t=	,(if v2000^n='b50t' then v2000^n'='v2000^v/ fi),
b40t=	,(if v2000^n='b40t' then v2000^n'='v2000^v/ fi),
b40u=	,(if v2000^n='b40u' then v2000^n'='v2000^v/ fi),
dupr=	,(if v2000^n='dupr' then v2000^n'='v2000^v/ fi),
dupl=	,(if v2000^n='dupl' then v2000^n'='v2000^v/ fi),
p1=	,(if v2000^n='p1' then v2000^n'='v2000^v/ fi),
p2=	,(if v2000^n='p2' then v2000^n'='v2000^v/ fi),
nowait	,(if v2000^n='nowait' then v2000^n/ fi),
now	,(if v2000^n='now' then v2000^n/ fi),
stderr=off	,(if v2000^n='stderr=off' then v2000^n/ fi),
mono	,(if v2000^n='mono' then v2000^n/ fi),
mast	,(if v2000^n='mast' then v2000^n/ fi),
full	,(if v2000^n='full' then v2000^n/ fi),
db=	,(if v2000^n='db' then v2000^n'='v2000^v/ fi),

Parámetro	Formato de extracción
dbn=	,(if v2000^n='dbn' then v2000^n'='v2000^v/ fi),
dict=	,(if v2000^n='dict' then v2000^n'='v2000^v/ fi),
k1=	,(if v2000^n='k1' then v2000^n'='v2000^v/ fi),
k2=	,(if v2000^n='k2' then v2000^n'='v2000^v/ fi),
cgi=	,(if v2000^n='cgi' then v2000^n'='v2000^v/ fi),
null	,(if v2000^n='null' then v2000^n/ fi),
tmp	,(if v2000^n='tmp' then v2000^n/ fi),
seq=	,(if v2000^n='seq' then v2000^n'='v2000^v/ fi),
iso=	,(if v2000^n='iso' then v2000^n'='v2000^v/ fi),
isotag1=	,(if v2000^n='isotag1' then v2000^n'='v2000^v/ fi),
bool=	,(if v2000^n='bool' then if v2000^v>'' then v2000^n'='v2000^v/ fi fi), if v2000:'^nbool^v' then 'bool=' (if v2000^n='bool' then if v2000^v>'' then v2000^v fi fi),/ fi,
mfbw	,(if v2000^n='mfbw' then v2000^n/ fi),
tbin=	,(if v2000^n='tbin' then v2000^n'='v2000^v/ fi),
tb=	,(if v2000^n='tb' then v2000^n'='v2000^v/ fi),
join=	,(if v2000^n='join' then v2000^n'='v2000^v/ fi),
jchk=	,(if v2000^n='jchk' then v2000^n'='v2000^v/ fi),
jch0=	,(if v2000^n='jch0' then v2000^n'='v2000^v/ fi),
jch1=	,(if v2000^n='jch1' then v2000^n'='v2000^v/ fi),
jmax=	,(if v2000^n='jmax' then v2000^n'='v2000^v/ fi),
jtag=	,(if v2000^n='jtag' then v2000^n'='v2000^v/ fi),
proc=	,(if v2000^n='proc' then v2000^n'='v2000^v/ fi),
convert=	,(if v2000^n='convert' then v2000^n'='v2000^v/ fi),
decod=	,(if v2000^n='decod' then v2000^n'='v2000^v/ fi),
gizp=	,(if v2000^n='gizp' then v2000^n'='v2000^v/ fi),
gizmo=	,(if v2000^n='gizmo' then v2000^n'='v2000^v/ fi),
giz1=	,(if v2000^n='giz1' then v2000^n'='v2000^v/ fi),
giz2=	,(if v2000^n='giz2' then v2000^n'='v2000^v/ fi),
putdir=	,(if v2000^n='putdir' then v2000^n'='v2000^v/ fi),
getdir=	,(if v2000^n='getdir' then v2000^n'='v2000^v/ fi),
sys=	,(if v2000^n='sys' then v2000^n'='v2000^v/ fi),
sys/show=	,(if v2000^n='sys/show' then v2000^n'='v2000^v/ fi),
text=	,(if v2000^n='text' then v2000^n'='v2000^v/ fi),
text/show=	,(if v2000^n='text/show' then v2000^n'='v2000^v/ fi),
lw=	,(if v2000^n='lw' then v2000^n'='v2000^v/ fi),
pft=	,(if v2000^n='pft' then v2000^n'='v2000^v/ fi),
invx=	,(if v2000^n='invx' then v2000^n'='v2000^v/ fi),
iso=	,(if v2000^n='iso' then v2000^n'='v2000^v/ fi),
outiso=	,(if v2000^n='outiso' then v2000^n'='v2000^v/ fi),
outisotag1=	,(if v2000^n='outisotag1' then v2000^n'='v2000^v/ fi),
fix=	,(if v2000^n='fix' then v2000^n'='v2000^v/ fi),
ln1=	,(if v2000^n='ln1' then v2000^n'='v2000^v/ fi),
ln2=	,(if v2000^n='ln2' then v2000^n'='v2000^v/ fi),
fst=	,(if v2000^n='fst' then v2000^n'='v2000^v/ fi),
fst/h=	,(if v2000^n='fst/h' then v2000^n'='v2000^v/ fi),
stw=	,(if v2000^n='stw' then v2000^n'='v2000^v/ fi),
ifupd=	,(if v2000^n='ifupd' then v2000^n'='v2000^v/ fi),

Parámetro	Formato de extracción
ifupd/create=	,(if v2000^n='ifupd/create' then v2000^n'='v2000^v/ fi),
ifupd/dict=	,(if v2000^n='ifupd/dict' then v2000^n'='v2000^v/ fi),
ifupd/create/dict=	,(if v2000^n='ifupd/create/dict' then v2000^n'='v2000^v/ fi),
fullinv=	,(if v2000^n='fullinv' then v2000^n'='v2000^v/ fi),
fullinv/dict=	,(if v2000^n='fullinv/dict' then v2000^n'='v2000^v/ fi),
fullinv/keep=	,(if v2000^n='fullinv/keep' then v2000^n'='v2000^v/ fi),
fullinv/m=	,(if v2000^n='fullinv/m' then v2000^n'='v2000^v/ fi),
fullinv/ansi=	,(if v2000^n='fullinv/ansi' then v2000^n'='v2000^v/ fi),
fullinv/dict/keep=	,(if v2000^n='fullinv/dict/keep' then v2000^n'='v2000^v/ fi),
fullinv/dict/m=	,(if v2000^n='fullinv/dict/m' then v2000^n'='v2000^v/ fi),
fullinv/dict/ansi=	,(if v2000^n='fullinv/dict/ansi' then v2000^n'='v2000^v/ fi),
fullinv/dict/m/ansi=	,(if v2000^n='fullinv/dict/m/ansi' then v2000^n'='v2000^v/ fi),
fullinv/ansi=	,(if v2000^n='fullinv/ansi' then v2000^n'='v2000^v/ fi),
fullinv/m/ansi=	,(if v2000^n='fullinv/m/ansi' then v2000^n'='v2000^v/ fi),
fullinv/keep/ansi=	,(if v2000^n='fullinv/keep/ansi' then v2000^n'='v2000^v/ fi),
fullinv/keep/m/ansi=	,(if v2000^n='fullinv/keep/m/ansi' then v2000^n'='v2000^v/ fi),
maxlk1=	,(if v2000^n='maxlk1' then v2000^n'='v2000^v/ fi),
maxlk2=	,(if v2000^n='maxlk2' then v2000^n'='v2000^v/ fi),
copy=	,(if v2000^n='copy' then v2000^n'='v2000^v/ fi),
append=	,(if v2000^n='append' then v2000^n'='v2000^v/ fi),
updatf=	,(if v2000^n='updatf' then v2000^n'='v2000^v/ fi),
create=	,(if v2000^n='create' then v2000^n'='v2000^v/ fi),
merge=	,(if v2000^n='merge' then v2000^n'='v2000^v/ fi),