Структура системы EC4\_BASE EC4\_CNFR EC4\_CNFW EC4\_G402 EC4\_SUPP EC4\_USER

Файлы конфигурации системы COLOR.CNF, EC.CNF, SYSTEM.CNF Файлы помощи Файл вопросов и сообщений системы

## Структура системы

После успешной инсталляции системы в директорию EC402 (возможно, у Вас это другая директория) должны образоваться следующие директории:

- EC4\_BASE Электронные каталоги системы.
  EC4\_CNFR Файлы конфигурации для пользователя с правами только на чтение.
  EC4\_CNFW Файлы конфигурации для пользователя с правами на чтение/запись.
  EC4\_G402 Документация к системе.
  EC4\_SUPP Файлы описания форматов экрана и документов, конверторы, словари и др. файлы, необходимые для корректной работы системы.
- EC4\_USER Директория пользователя.

## Директория EC4\_BASE

В этой директории после установки создаются еще 7 поддиректорий, количество поддиректорий соответствует количеству установленных электронных каталогов. В каждой такой поддиректории будет накапливаться библиографическая информация (записи ЭК) для конкретного каталога. В директории BIBL будут записи каталога Статьи, в директории BOOK - записи каталога Книги, в директории СОМР записи каталога Комплектование, в директории LMSU - записи каталога НБ МГУ, в директории PERI - записи каталога Периодика, в директории PODP - записи каталога Подписка, в директории READ - записи каталога Читатели.

Имена директорий соответствуют латинскому имени каталога, которое задается в файле конфигурации каждого каталога.

NB! Обратите внимание на название поддиректорий; далее везде, где они будут, встречаться будет означать, что информация относится, например, к каталогу Книги, если это поддиректория BOOK, и это не обязательно в директории EC4\_BASE.

Остановимся на вопросе: какие файлы будут образовываться в директории каждого из каталогов и какая у них структура.

В директории будут создаваться следующие файлы, которым даны условные, а не реальные названия:

МС МЕМО Поисковые файлы

MC файлы. В этих файлах хранятся библиографические записи любого каталога в закодированном виде, если быть точнее, то в формате USMARC. Этих файлов будет много, и у них будут разные имена, например, MC4.DBF, MC5.DBF, MC12.DBF. Система создает их сама по мере необходимости.

Структура файла MC4.DBF

Имя поля	Тип поля	Длина поля
LCN	Numeric	10
MF1	Character	128
MF2	Character	128
MF3	Character	128
MF4	Character	128

Структура файла MC5.DBF

Имя поля	Тип поля	Длина поля
LCN	Numeric	10
MF1	Character	128
MF2	Character	128
MF3	Character	128
MF4	Character	128
MF5	Character	128

Как видно, оба файла практически имеют одинаковую структуру, только файл MC5.DBF на одно поле длинее, что, собственно, и отражено в его названии. Так, файл MC12.DBF будет иметь поля с MF1 по MF12.

В первое поле во всех файлах заносится порядковый номер записи в данном файле, далее идет сама "марковская" запись. Если ее длина более 256 байт (128\*2), но менее 384 байт (128\*3), то она будет записана в файл MC3.DBF. Если длина более 384 байт, но менее 512 байт (128\*4), то в файл MC4.DBF, и т.д.

Для каждого из MC файлов для ускорения доступа к данным создается индексный файл с таким же именем, но расширением NTX.

Файл MEMO.DBF. Это очень важный файл. Он осуществляет координацию между MC файлами. Его структура:

Имя поля	Тип поля	Длина поля
CN	Numeric	10
MC_NB	Numeric	3
MC_NR	Numeric	10

В первое поле заносится порядковый номер записи по каталогу, во второе поле - номер MC файла, где хранится запись, в последнее поле - номер записи в самом MC файле.

Для МЕМО файла создается также индексный файл - MEMO.NTX.

Поисковые файлы. Для каждого поискового элемента создается поисковый файл. Имя и параметры этого файла задаются в файле ELECAT.DBF (см. главу Электронный каталог. Файлы конфигурации ЭК). Имя файла должно быть уникальным и состоять из трех латинских букв. Версия 4.02 поддерживает ранее принятые соглашения об именах поисковых таблиц, однако в дальнейшем это может измениться.

Первое поле у поискового файла всегда называется CN, тип - Numeric, длина - 10. Второе поле называется по имени файла (которое Вы можете задать сами), тип - Character, длина - какую зададите сами.

В первое поле заносится номер записи в ЭК, во второе поле - значение поискового элемента. Если это файл авторов, то будут занесены все авторы, которые находятся в этой записи ЭК.

Для каждой поисковой таблицы создается два индексных файла. Имя первого файла образуется из первых трех символов имени

поискового файла, а далее идет \_CN и расширение NTX. Этот индексный файл проиндексирован по первому поля для быстрого поиска по номеру записи. У второго индексного файла имя такое же, как и у файла с расширением DBF. Длину индекса второго файла Вы можете задать сами. Об этом смотрите в главе Настройка.

NB! Например, если у Вас есть поисковый элемент - автору, то для него в программе будут созданы 3 файла - AUT.DBF, AUT.NTX, AUT\_CN.NTX.

## Директория EC4\_CNFR

В директории создаются 4 поддиректории для каждого каталога, доступного пользователю с правами только на чтение. Это поддиректории: BIBL, BOOK, LMSU, READ. В каждой из них содержатся файлы, описывающие конкретный каталог для пользователя с правами только на чтение. О файлах конфигурации каталога смотрите главу Электронный каталог. Но еще в самой директории есть 3 файла, относящиеся к конфигурации всей системы в целом. Это файлы:

COLOR.CNF	Файл,	задающий	комбинацию	цветов	для	меню	системы.	
EC.CNF	Файл,	задающий	список кат	алогов,	дост	гупный	í	
	пользователю.							
SYSTEM.CNF	Файл,	задающий	параметры	системы				

# Директория EC4\_CNFW

В директории создаются еще 7 поддиректорий для каждого каталога, где содержатся файлы конфигурации каждого каталога. О файлах конфигурации каталога смотрите главу Электронный каталог. Но еще в самой директории есть 3 файла, относящиеся к конфигурации всей системы в целом. Это файлы:

COLOR.CNF	Файл,	задающий	комбинацию	цветов	для	меню	системы.	
EC.CNF	Файл,	задающий	список кат	алогов,	досл	упный	í	
	пользователю.							
SYSTEM.CNF	Файл,	задающий	параметры	системы				

#### Директория EC4\_G402

-----

Директория документации к системе. Состоит из следующих поддиректорий:

EC402_GS	Описание установки системы и начало работы.
EC402_TR	Структура и настройка системы. Файлы описания
	экрана, документов, клавиш и файлы перекодировки.
	Приложения.
EC402_TS	Описание технологических процессов.
EC402_UG	Руководство пользователя.

Директория EC4\_SUPP

Состоит из следующих поддиректорий:

 Поддиректория
 Что в поддиректории находится

 BLC
 Файлы для занесения информации в поисковые таблицы

 и файлы дополнительной обработки информации перед
 проведением поиска.

CNV	Файлы перекодировки.
DIC	Словари.
DOC	Файлы документов.
FMT	Файлы формата экрана.
KEY	Файлы, описывающие работу отдельных клавиш.

Эти поддиректории определяют работу всех каталогов. Более подробно о файлах в каждой из этих поддиректорий будет рассказано далее в этом руководстве.

Также в директории содержатся файлы:

HELP.DBF -	
HELP.DBT -	Файлы помощи системы.
HELP.NTX -	
QUESTION.DBF	Файл вопросов и сообщений системы.
EC4.EXE	Файл, содержащий код программы.
EC4E.EXE	Файл, содержащий код программы для компьютеров с
	расширенной памятью.

Поддиректория BLC

Директория содержит файлы:

- tin.blc Файл обработки инвентарных номеров перед занесением в поисковую таблицу инвент.номеру. Если в поле Инвентарный номер введено число, то оно будет дополнено 0 слева до 10 знаков, если введено УК, то будет дополнено пробелами.
- tsu.blc Все то же самое, что написано выше, только для поисковой таблицы номеру КСУ.
- tin.get Данные, введенные пользователем, дополняются 0 или пробелами перед проведением поиска по поисковому элементу инвент.номеру.
- tsu.get Все то же самое, что написано выше, только для поискового элемента номеру КСУ.

Файлы, указанные ниже, используются только при работе с каталогом МГУ. Некоторые из них достаточно полезны, как, например, файлы для названия или серии, и Вы их можете подключить к поисковым таблицам своих каталогов.

- tca.blc Таблица занесения каталожных индексов, чтобы они не разбивались. Поисковый элемент - индексам НБ МГУ.
- til.blc Таблица занесения инвентарных номеров, как они есть: год, номер путевки, порядковый номер. Поисковый элемент - инв.номерам.
- tse.blc Таблица занесения заглавия серии в соответствии со значением поля Незн.символы. Поисковый элемент - серии.
- tti.blc Таблица занесения информации из поля 245а с учетом параллельного заглавия и незначащих символов, а также занесение информации из полей 242а, 247а, 100t, 100p. Поисковый элемент названию.

tca.get, til.get, tse.get, tti.get файлы ввода информации и проведения поиска по выше перечисленным поисковым элементам.

## Поддиректория CNV

Поддиректория содержит файлы:

- compl.con Для удаления следующих меток: 093a, 093b, 583a, 583c, 583b, 583n, 583k, 583d, 583l, 583x. Используется при копировании записей из каталога Комплектование в каталог Книги с использованием файла перекодировки.
- gpntb.con Для сохранения записей каталога в формате UNIMARC библиотеки ГПНТБ. Файл используется в режиме Сохранение информации в файле с подключением файла перекодировки.
- ldb\_ld1.con Конвертирует записи из системы АИБС "МАРК" в формат USMARC программы Библиотека 4.02. Используется в режиме Загрузка информации из файла с использованием файла перекодировки.
- new.con Конвертирует записи из системы Библиотека 3.02 в формат USMARC программы Библиотека 4.02. Используется в режиме Перекодировка информации по каталогу.
- newload.con Конвертирует записи из системы Библиотека 3.02 в формат USMARC программы Библиотека 4.02. Используется в режиме Загрузка информации из файла с использованием файла перекодировки.

Поддиректория DIC

Подиректория содержит файлы:

d_092a.dbf -+ d_092a1.ntx   d_092a2.ntx -+	Файлы словаря рубрик ГАСНТИ. Используются при занесении библиографической информации, при поиске, а также при создании бюллетеня.
d_952a.dbf -+ d_952a1.ntx   d_952a2.ntx -+	Файлы словаря Тип литературы. Используются при занесении библиографической информации, а также при создании документа Лист книги суммарного учета.
keyall.dbf -+ keyall1.ntx   keyall2.ntx -+	Файлы словаря ключевых слов по следующим областям знаний: Биология, География, Геология, Государство и право, История, Литературоведение, Математика, Медицина, Науковедение, Педагогика, Психология, Религия, Физика, Химия, Экономика, Электроника, Языкознание.

Поддиректория DOC

Поддиректория содержит файлы:

b01g_42.asf	Управляет процессом создания заголовков в бюллетене.
b01g_42.doc	Файл описания документа Бюллетень.
i001_42.doc	Файл описания документа Инвентарная книга.
i002_42.doc	Файл описания документа Лист книги суммарного

учета.

- 1001\_42.doc Файл описания документа Список литературы в каталогах: Книги, Статьи, Книги\_МГУ
- 1002\_42.doc Файл описания документа Список литературы в каталоге Комплектование.
- r001\_42.doc Файл описания документа Список читателей.
- r002\_42.doc Файл описания документа Список задолжников.
- sp01\_42.doc Файл описания документа Форма СП-1 (по названию). По названию означает, что сортировка записей в документе будет производиться по названию издания.
- sp02\_42.doc Файл описания документа Форма СП-1 (по индексу). По индексу означает, что сортировка записей в документе будет производиться по индексу РП издания.
- sp07\_42.doc Файл описания документа Форма СП-7 (по названию).
- sp08\_42.doc Файл описания документа Форма СП-7 (по индексу).
- spl1\_42.doc Файл описания документа Рабочая форма (по названию).
- sp12\_42.doc Файл описания документа Рабочая форма (по индексу).
- sp17\_42.doc Файл описания документа Сводная ведомость (по названию).
- sp18\_42.doc Файл описания документа Сводная ведомость (по индексу).

Поддиректория FMT

Поддиректория содержит файлы:

a01_42.fmt	Полная карточка (v4.2)
a02_42.fmt	Все поля
a03_42.fmt	Сокращенная карточка (v4.2)
a04_42.fmt	Карточка с длинным шифром (v4.2)
b01_42.fmt	Оборот карточки (v4.2)
c01_42.fmt	Карточка для комплектования (v4.2)
c02_42.fmt	Оборот карточки (Комплектование)
f01_42.fmt	Формуляр (v4.2)
r01_42.fmt	Читательское требование
r02_42.fmt	Карточка для читателя

Поддиректория КЕҮ

Поддиректория содержит файлы:

isbnchek.blc Предназначен для проверки дублетности.

- setall.blc Вызов меню Общие установки. Клавиши Alt+F8 из Основного меню системы.
- setlink.blc Устанавливает связь со связанным каталогом. По умолчанию - с каталогом Читатели. Клавиши Alt+F2 из Основного меню системы.

- showall.blc Показывает все записи связанного каталога. Клавиши Alt+F3 из Основного меню системы.
- showlink.blc Показывает только связанные с текущей записью записи из связанного каталога. Клавиши Alt+F4 из Основного меню системы.
- textquir.blc Позволяет вводить запрос на основе булевой алгебры. Клавиши Alt+F7 из Основного меню системы.
- writebuf.blc Осуществляет запись заданного множества записей в буфер. Клавиши Alt+F5 из Основного меню системы.

## Директория EC4\_USER

-----

Состоит из следующих поддиректорий:

RDR_ONLY	Поддиректория	для	работы	пользователя	С	правами	
	только на чтен	ие.					
RDR_WRT	Поддиректория чтение/запись.	для	работы	пользователя	С	правами	на

В каждой из этих директорий находится два файла: ec4.bat и ec4e.bat.

Первый предназначен для запуска обычной программы, второй - запуск программы для расширенной памяти.

Содержание файла ec4.bat для пользователя с правами на чтение/запись:

c:\ec402\ec4\_supp\ec4 \ec4\_cnfw\ec.cnf /n

полное имя файла	полное имя	ключ, который запрещает
для запуска	файла со списком	системе проверку наличия
программы	доступных каталогов	строки SET CLIPPER=F85
		в файле AUTOEXEC.BAT

Содержание файла ec4e.bat для пользователя с правами на чтение/запись:

c:\ec402\ec4\_supp\ec4e \ec4\_cnfw\ec.cnf /n

Файлы конфигурации системы

Файл COLOR.CNF

Содержание файла после инсталляции:

01.	MarcEdit.Информационная строка	{"W/B","W/N","N/W"}
02.	MarcEdit.Tekct	{"W/B,+GR/B,n/W","W/N,+W/N,n/W",
		"N/W,N/W,W/n"}
03.	MarcEdit.Заголовки	{"+GR/B","+W/N","N/W"}
04.	Клавишы. Цифры	{ "+W/N", "+W/N", "N/W" }
05.	Клавишы. Буквы	{"n/*bg","N/W","W/N"}
06.	Помощь	{"b/w","N/W","W/N"}
07.	Строка подсказки. Начало	{ "+w/N", "+W/N", "N/W" }
08.	Строка подсказки. Конец	{"n/*w","N/W","W/N"}
09.	Редактор. Кодируемые данные	{"N/BG,+W/B","N/W,+W/N","W/N,N/W"}
10.	Рабочие листы. Текст	{"G/N","W/N","N/W"}
11.	Рабочие листы. Данные	{ "+W/N", "+W/N", "N/W" }

12. Рабочие листы. Клавишы 13. Рабочие листы. Клавишы 14. Рабочие листы. Рамка окна 14. Рабочие листы. Файловое окно 15. Установка. Текст 16. Установка. Данные 17. Установка. Клавишы 18. Диалог 1. Текст 19. Диалог 1. Заголовок 20. Диалог 1. Не выбрано 21. Диалог 1. Выбрано 22. Диалог 1. Тень 23. Диалог 1. Фон 24. Выбор. Текст 25. Выбор. Заголовок 26. Рабочие листы. Сообщения 27. Редактор. Отмеченный блок 28. Экран. Текст 29. Экран. Имена окон 30. Экран. Активное окно 31. Экран. Название каталога 32. String 23-24 33. Диалог 2. Текст 34. Диалог 2. Тень 35. Индикация. Столбик 36. Ошибки 37. Поиск. Ввод запроса 38. Настройка. Текст 39. Настройка. Заголовки 40. Пароль 41. Поиск. Восстановление запроса 42. Ввод данных 43. Меню. Текст 44. Меню. Заголовки 45. Меню. Не выбрано 46. Меню. Выбрано 47. Меню. Тень 48. Меню. Фон 49. Индикация. Текст 50. Индикация. Заголовки 51. MarcEdit.Инф.строка. Работа. 52. Редактор текст 53. Верт.меню.Текст 54. Верт.меню.Заголовок 55. Редактор инф.строка 56. Нов.список б.рамка и текст 57. Нов.список не выд.м.рамка(текст) { "+n/w,+w/rb", "n/w,+w/n", "w/n,n/w" } 58. Нов.список выд. м.рамка 59. Нов.список Get 60. Нов.список действие 61. Нов.список клавиши 62. Нов.список тень 63. Словарь. Текст 64. Error. Text 65. Error. Select 66. Error. Not-select 67. Error. Shadow 68. Message 69. Error. Bright(Pr) 70. Yes\_no first sel. letter

{"W/N", "W/N", "N/W"} {"+R/N", "+W/N", "N/W"} { "G/N,N/W,,,G/N", "W/N,N/W,,,W/N", "N/W,W/N,,,N/W" } {"G/N", "W/N", "W/N"} {"+R/N,N/W,,,+R/N","+W/N,N/W,,,+W/N", "N/W,W/N,,,N/W"} {"RB/N", "W/N", "N/W"} {"n/BG\*","N/W","W/N"} {"+r/BG\*", "N/W", "W/N"} {"N/W", "N/W", "W/N"} { "b/\*W", "+W/N", "N/W" } { "N/\*BG", "N/W", "W/N" } {"BG/B", "W/N", "W/N"} { "N/W, +W/N", "N/W, +W/N", "W/N, N/W" } {"B/W", "N/W", "W/N"} {"G/N", "W/N", "N/W"} {"N/W", "N/W", "W/N"} { "W/B,N/W", "W/N,N/W", "N/W,W/N" } {"N/BG", "N/W", "W/N"} {"+W/B,+W/B","+W/N,+W/N","N/W,N/W"} { "+W/B,+W/B", "+W/N,+W/N", "N/W,N/W" } {"W/N","W/N","N/W"} { "N/BG,+W/N", "N/W,+W/N", "W/N,N/W" } {"N/N", "N/N", "W/W"} { "N/W,N/W", "N/W,N/W", "W/N,W/N" } { "+W/R,W/N", "+W/N,N/W", "N/W,W/N" } {"N/BG,+W/B","N/W,+W/N","W/N,N/W"} { "+W/BG,+W/BG", "N/W,N/W", "W/N,W/N" } {"+GR/BG", "N/W", "W/N"} { "+w/BG, B/B", "N/W, N/N", "W/N, W/W" } { "N/W, +W/N", "N/W, +W/N", "W/N, N/W" } {"G/N,G/N", "N/W,N/W", "W/N,W/N"} {"+W/BG", "N/W", "W/N"} {"+GR/BG", "N/W", "W/N"} { "N/W", "N/W", "W/N" } {"+W/B","+W/N","N/W"} {"N/BG", "N/W", "W/N"} {"BG/B", "W/N", "W/N"} { "+W/BG,+W/BG", "N/W,N/W", "W/N,W/N" } {"+GR/BG", "N/W", "W/N"} {"N/W", "N/W", "W/N"} { "w/b,+w/b,n/w,+gr/b", "w/n,+w/n, n/w,+w/n", "n/w,n/w,w/n,w/n" } { "n/w, +w/n, +n/w, ,n/w", "n/w, +w/n, n/w, ,n/w","w/n,n/w,w/n,,w/n"} {"b/w","n/w","w/n"} {"w/b", "w/n", "n/w"} {"n/w","n/w","w/n"} { "+w/w,+w/rb", "n/w,+w/n", "w/n,n/w" } {"b/w,+w/rb,,,b/w","n/w,+w/n,,,n/w", "+n/w,+w/rb,,,+n/w"} {"n/bg","n/w","w/n"} {"+w/bg", "n/w", "+n/w"} {"n/w","n/w","w/n"} { "N/BG,N/W", "W/N,N/W", "W/N,N/W" } {"+w/r", "n/w", "w/n"} {"+w/b","+w/n","n/w"} {"n/w","n/w","w/n"} {"n/r","n/w","w/n"} {"+w/w","n/w","w/n"} {"+Gr/r","+w/n","n/w"} {"r/\*w","+w/n","w/n"}

71.	Yes_no first dissel. let	ter {"r/w","n/w","w/n"}
72.	Choice Letter	{"gr/w","n/w","w/n"}
73.	Button ( Select )	{"+w/rb","+w/n", "n/w" }
74.	"EDT" and "HLP",text,blo	ck {"+n/w,,n/bg","+n/w,,w/n","w/n"}

NB! Реально строки в файле не разбиваются, здесь перенос на следующую строку сделан только для наглядности.

Файлы одинаковые, что для пользователя с правами только на чтение, что для пользователя с правами на чтение/запись. Но если у Вас есть желание, то они могут стать разными.

Файл делался так давно, что трудно сказать, к какому конкректному меню относятся все эти пункты. Но мысль была такая, что файл состоит из трех колонок. В первой колонке код меню, который и используется непосредственно в программе, вторая колонка - описание меню (что-то типа комментария), третья колонка - задание цвета.

Вероятнее всего, в программе используются не все элементы, но какие точно используются, сказать уже трудно.

Структура третьей колонки: первое значение в кавычках относится к цветному монитору, второе - к черно-белому, третье - к черно-белому Notebook.

Если Вы хотите поменять цвет, то нужно менять значения в последней колонке. Далее прилагается список возможных цветов.

Цвет	Цветной монитор	Монохромный монитор
N	Черный	Черный
В	Синий	Подчеркивание
G	Зеленый	Белый
BG	Голубой	Белый
R	Красный	Белый
RB	Сиреневый	Белый
GR	Оранжево-коричневый	Белый
W	Белый	Белый
+N	Серый	Черный
+B	Ярко-синий	Подчеркивание яркое
+G	Ярко-зеленый	Ярко-белый
+BG	Ярко-голубой	Ярко-белый
+R	Ярко-красный	Ярко-белый
+RB	Ярко-сиреневый	Ярко-белый
+GR	Желтый	Ярко-белый
+W	Ярко-белый	Ярко-белый

Файл EC.CNF

Содержание файла после установки системы для пользователя с правами только на чтение:

C:\EC402\EC4\_CNFR\BOOK\BOOK.CNF Книги C:\EC402\EC4\_CNFR\BIBL\BIBL.CNF Статьи C:\EC402\EC4\_CNFR\PERI\PERI.CNF Периодика C:\EC402\EC4\_CNFR\LMSU\LMSU.CNF Книги\_МГУ

Содержание файла для пользователя с правами на чтение/запись:

C:\EC402\EC4\_CNFW\BOOK\BOOK.CNF Книги C:\EC402\EC4\_CNFW\BIBL\BIBL.CNF Статьи C:\EC402\EC4\_CNFW\COMP\COMP.CNF Комплектование C:\EC402\EC4\_CNFW\PERI\PERI.CNF Периодика C:\EC402\EC4\_CNFW\PODP\PODP.CNF Подписка C:\EC402\EC4\_CNFW\READ\READ.CNF Читатели C:\EC402\EC4\_CNFW\LMSU\LMSU.CNF Книги\_МГУ

В файле содержится список подключенных каталогов к системе с указанием пути и самого файла конфигурации для каждого из каталогов, а также название каталога.

C:\EC402\EC4_CNFW\LMSU\	LMSU.CNF	Книги_МГУ	
Путь	Файл конфигу- рации каталога	Название	каталога

NB! В версии 4.02 принято соглашение, что файл конфигурации называется по имени директории каталога и имеет расширение CNF.

Файл SYSTEM.CNF

Содержание файла для пользователя с правами только на чтение:

WRT OFF
QUE C:\ec402\ec4\_supp\question.dbf
HLP C:\ec402\ec4\_supp\help.dbf
QMC 5
QEC 5
NAL 1
MON COLOR

Содержание файла для пользователя с правами на чтение/запись:

WRT ON QUE C:\ec402\ec4\_supp\question.dbf HLP C:\ec402\ec4\_supp\help.dbf QMC 5 QEC 5 NAL 1 MON COLOR

Файлы отличаются только первой строкой. В первом случае запрещена запись в файл помощи (признак OFF), во втором случае запись разрешена (признак ON).

Далее идет указание пути и имени файла вопросов и сообщений системы, затем задается путь и имя файла помощи системы.

Параметр QMC задает, сколько одновременно открывать файлов MC (файлы -, в которых непосредственно хранятся записи каталога). При задании этого параметра надо учитывать, что малое или очень большое число замедляет работу системы. Надо соблюдать разумную середину.

Параметр QEC задает, сколько одновременно открывать поисковых таблиц. Ориентируйтесь на среднее значение.

Параметр NAL задает сколько раз программа будет пытаться открыть файл в сети, если он занят другим пользователем.

Последней строкой задается тип монитора. Может быть:

COLOR

MONO LAPTOP

Файлы помощи \_\_\_\_\_

> HELP.DBF — HELP.DBT — Файлы помощи системы. HELP.NTX —

Структура файла HELP.DBF:

Имя поля	Тип поля	Длина поля
KEY	Character	10
HELP	Memo	10

HELP.NTX - файл индексов для файла HELP.DBF HELP.DBT - текст помощи.

NB! Не надо трогать эти файлы. Любые их изменения без программы, скорее всего, повлекут невозможность вызова текста помощи на экран.

Файл вопросов и сообщений системы

Структура файла QUESTION.DBF:

Имя поля	Тип поля	Длина поля
NQ	Numeric	4
QUESTION	Character	76

В первое поле заносится код сообщения, во второе - его значение, которое Вы и видите на экране.

NB! Не вносите никаких изменений для дальнейшей корректной работы системы.

Глава 2: Электронный каталог. Файлы конфигурации ЭК

Электронный каталог

Файлы конфигурации электронного каталога						
BOOK.CNF	файл конфигурации ЭК					
ELECAT.DBF	описание поисковых таблиц и задание путей к файлам					
	FIELD.DBF и USER.DBF					
FIELD.DBF	описание полей электронного каталога					
USER.DBF	файл паролей					
LEVEL.CNF	файл библиографических уровней					
KEY.CNF	файл "внешних" функций					
DOCUMENT.CNF	список файлов описания документов					
CONVERT.CNF	список файлов перекодировки (конверторов)					
ADDCARD.CNF	список возможных добавочных описаний					

## Электронный каталог

Система может работать с несколькими электронными каталогами. Каждый электронный каталог имеет:

Уникальное имя Свой файл конфигурации Свои поисковые таблицы Свой набор заполняемых полей Свой файл паролей Файл библиографических уровней Файл библиографических уровней Файл "внешних" функций Список файлов описания документов Список файлов перекодировки (конверторов) Список возможных добавочных описаний

NB! Возможен более частный случай, когда один и тот же файл используется в нескольких каталогах. Все зависит от того, как Вы настроите систему.

Каждый ЭК имеет уникальное имя. В версии 4.02 имя каталога должно состоять из четырех латинских букв в верхнем регистре. Например:

Правильные имена: ВООК или READ или MSUB НЕправильные имена: ВОО1 или LIB или ЛИБК или READER

NB! Версия 4.02 поддерживает работу с именами электронных каталогов, принятых в более ранних версиях. Однако мы не рекомендуем создавать новые каталоги, не учитывая новых требований.

Имя каталога задается в файле конфигурации этого каталога.

Каждый электронный каталог описывается файлом конфигурации. В версии 4.02 принято соглашение, что файл конфигурации называется по имени электронного каталога и имеет расширение CNF. Например:

Имя каталога	Имя файла конфигурации
BOOK	BOOK.CNF
BIBL	BIBL.CNF
READ	READ.CNF

В файле конфигурации размещается информация о настройке данного электронного каталога.

Список ЭК, доступных данному пользователю, по умолчанию задается в файле EC.CNF. Но таких файлов может быть много разных и имя файла указывается в каждом конкретном случае особо при запуске программы. Описание содержания и структуры файла EC.CNF смотрите в главе Структура системы.Файлы конфигурации системы.

Информация из всех электронных каталогов размещается в директории EC4\_BASE. В директории находятся поддиректории, имена которых совпадают с именами электронных каталогов. В этих поддиректориях размещаются файлы с информацией соответствующего электронного каталога, а также поисковые таблицы ЭК.

NB! Меняя настройки в файле конфигурации электронного каталога, можно настроить его на другое месторасположение.

Каждый электронный каталог описывается несколькими файлами, содержащимися в директории конфигурации электронного каталога. Все директории конфигураций размещаются либо в директории EC4\_CNFR, либо в директории EC4\_CNFW.

Одному электронному каталогу может соответствовать несколько директорий конфигурации. Например:

Имя каталога Имя директории конфигурации

ВООК \EC4\_CNFR\BOOK - конфигурация для доступа только на чтение \EC4\_CNFW\BOOK - конфигурация для доступа на чтение/запись

В директории конфигурации размещаются файлы, задающие настройки электронного каталога. Рассмотрим в качестве примера директорию \EC4\_CNFW\BOOK.

В директории \EC4\_CNFW\BOOK находятся следующие файлы:

BOOK.CNF	-	файл конфигурации ЭК ВООК
ELECAT.DBF	-	пути и описание поисковых таблиц
FIELD.DBF -	+	файлы, задающие описание полей
FIELD.DBT		и подполей ЭК
FIEI1.NTX		
FIEI2.NTX -	-+	
USER.DBF	-	файл паролей для доступа к ЭК
LEVEL.CNF	-	список библиографических уровней
KEY.CNF	-	список "внешних" функций
DOCUMENT.CNF	-	список файлов описания документов
CONVERT.CNF	-	список файлов перекодировки (конверторов)
ADDCARD.CNF	-	список возможных добавочных описаний

Эти файлы задают основные настройки ЭК.

Ниже приведено содержание файла BOOK. CNF с расширенными комментариями и пояснениями. Символ \* означает, что строка является комментарием. \* Разрешение/запрещение записи. \* ON - разрешает ввод и редактирование информации в ЭК. \* OFF - запрещает ввод и редактирование информации в ЭК. WRT ON \* Имя каталога. \* Задает уникальное имя электронного каталога. \* 4 латинских буквы в верхнем регистре. TEC BOOK \* Подстановка текущего диска. \* Если в файлах конфигурации в "пути" диск не указан, то происходит \* подстановка имени диска, указанного ниже. \* ? - запрещает подстановку. \* Пример: DSK N KEY \EC4\_CNFW\BOOK\KEY.CNF \* PKB C:\EC4\_SUPP\KEY \* В результате система будет работать со списком внешних \* функций, расположенным в файле N:\EC4\_CNFW\BOOK\KEY.CNF, \* сами же файлы внешних функций будут разыскиваться в \* директории C:\EC4 SUPP\KEY. DSK C:\EC402 \* Конфигурация внешних клавиш. \* Задает файл, содержащий список клавиш и соответствующие \* им файлы блоков кода, для выполнения "внешних" функций. KEY \EC4\_CNFW\BOOK\KEY.CNF \* Путь к размещению файлов блоков кода для "внешних" функций. PKB \EC4 SUPP\KEY \* Всегда SHARED NUS SHARED \* Управляет списком доступных утилит в режиме сервиса, клавиша F2 \* в окне Каталог Основного меню системы. \* ON - дает возможность только сохранять информацию. \* OFF - дает все возможности сервиса. NET OFF \* Задает путь к файлу ELECAT.DBF, в котором с 1 по 10 записи \* указаны пути и имена файлов системной конфигурации, а с 11 \* записи и далее - описание поисковых таблиц. PRF \EC4\_CNFW\BOOK\ELECAT.DBF \* Директория размещения файлов блоков кода для занесения \* нестандартных значений в поисковые таблицы (файлы - TXX.BLC, \* где Х - цифра или буква) и файлов обработки данных запроса для \* нестандартных поисковых таблиц ТХХ.GET (где X - цифра или буква) PBL \EC4 SUPP\BLC

* Задается путь к файлу библі PLE \EC4_CNFW\BOOK	иографических уровней - LEVEL.CNF
* Умолчание для библиографич BBL m	еского уровня
* Путь к файлу DOCUMENT.CNF PDO \EC4_CNFW\BOOK	
* Путь к файлу ADDCARD.CNF PAD \EC4_CNFW\BOOK	
* Путь к файлу CONVERT.CNF PCO \EC4_CNFW\BOOK	
* Путь к файлу ECCONECT.CNF PEC \EC4_CNFW\BOOK	
* Форматы экрана * Директория, где размещаютс: PFT \EC4_SUPP\FMT	я файлы описания форматов экрана
<ul> <li>* называемое "кольцо".</li> <li>* А01_42 - полная карточка</li> <li>* А02_42 - полный список поле</li> <li>* А03_42 - сокращенная карточ</li> <li>* А04_42 - карточка с длинны</li> <li>* B01_42 - оборот карточки</li> <li>* F01_42 - формуляр</li> <li>* R01_42 - читательское требо</li> <li>NFS А01_42, А02_42, А03_42, А04</li> <li>* Связь каталогов</li> <li>* Задана связь с каталогом чи LNK \EC4_CNFW\READ\READ.CNF 1</li> </ul>	ей чка м шифром ование _42,801_42,F01_42,R01_42 итатели NBO NRC-15 REA-25 Читатели
Файл конфигурации связанного каталога	Название связаннной таблицы Задано имя поисковой таблицы,из которой будут показываться значения для второй колонки, а также задано количество символов для показа. Задано имя поисковой таблицы, из которой будут показываться значения для первой колонки, а также задано количество символов для показа.
	Гоисковая таблица, по которой происходит связь. В данном случае это поле – Код экземпляра. В связанном каталоге обязательно должна быть поисковая таблица по этому полю.

Для понимания функционирования программы приведем пример того, как программа выполняет создание документа. Пусть в файле конфигурации задано:

DSK N PDO \EC4\_CNFW\BOOK

тогда при входе в меню Документы программа сначала прочитает список файлов описания документов из файла:

N:\EC4\_CNFW\BOOK\DOCUMENT.CNF

Пример этого списка:

C:\EC4\_SUPP\L001\_42.DOC Список литературы C:\EC4\_SUPP\DOC\B01G\_42.DOC Бюллетень \DOC\I001\_42.DOC Инвентарная книга \DOC\I002\_42.DOC Лист книги суммарного учета

В две последние строчки будет произведена подстановка диска N. Окончательно программа будет работать со следующим списком файлов описания документов:

C:\EC4\_SUPP\L001\_42.DOC Список литературы C:\EC4\_SUPP\DOC\B01G\_42.DOC Бюллетень N:\DOC\I001\_42.DOC Инвентарная книга N:\DOC\I002\_42.DOC Лист книги суммарного учета Файл ELECAT.DBF содержит путь к словарям, к файлам FIELD.DBF и USER.DBF, а также список поисковых элементов каталога (с путями, именами и другими параметрами).

Внести изменения в файл можно из программы (что предпочтительнее), а также с помощью любого редактора DBF файлов (например, DBU.EXE).

- NB! Файл ELECAT.DBF не обязательно должен находиться в директории \EC4\_CNFW\BOOK. Его имя задается в файле конфигурации каталога, например, BOOK.CNF после описателя PRF.
- NB! Имя файла, указанного в файле BOOK.CNF после описателя PRF, должно быть ELECAT.DBF. Может меняться только путь к нему.

Ниже приводится пример файла ELECAT.DBF для каталога Книги с комментариями. Первая колонка представляет номер записи и в сам файл не входит. Строки, расположенные выше номеров записей, обозначают имя поля и его тип (как никак, ELECAT.DBF - это файл базы данных, а не просто текстовый).

		FILE_						
	PATH	NAME	NTX	TYPE	TITLE	CODE	TAG1	SF1
	"C"	"C"	"C"	"C"	"C"	"N"	"C"	"C"
1						0		
2	\EC4_BASE\BOOK\	MEMO	MEMO	DBF		0		
3						0		
4	\EC4_CNFW\BOOK\	FIELD	FIELD	DBF		0		
5						0		
6		_	_			0		
./	\EC4_CNFW\BOOK\	USER	USER	DBF		0		
8	\EC4_SUPP\DIC\					0		
10						0		
10		3 T T CT	3 T T CT	0		0	100	_
	\EC4_BASE\BOOK\	AUT	AUT	0	автору	∠U ⊑0	100	a
12	\EC4_BASE\BOOK\	TTT	TTT	0	заглавию	50	245	a
13	\EC4_BASE\BOOK\	COR	COR	0	коллек.автору	30	110	a
14	\EC4_BASE\BOOK\	SER	SER	0	названию серии	30	490	a
15	\EC4_BASE\BOOK\	TER	TER	0	предмет.руорика	M 25	650	a
17	\EC4_BASE\BOOK\	KEY	KEY	0	ключевым словам	∠5	000	a
1 /	\EC4_BASE\BOOK\	GAS	GAS	0	индексам ГАСНТИ	10	092	a
10	\EC4_BASE\BOOK\	UDC	UDC	0	индексам удк	10	080	a
19	\EC4_BASE\BOOK\	LCC	LCC	0	индексам ББК	11	091	d
∠U 21	LEC4_BASE BOOK	TURA	TURA	0	индексам ISBN	10	UZU TTN	d
⊿⊥ วว	LC4_DASE \BOOK \			0	инвент.номеру	0 T 0	0E0	h
∠∠ วว	LEC4_BASE BOOK	SIG	SIG	0	месту хранения	10	054 260	D h
22	LC4_DASE \BOOK \	PUD VEN	PUD VEN	0	издательству	10	200	u c
24	LC4_DASE (BOOK )			0	году издания	10	200	+
25	LCT_DASE (BOOK (			0	коду экземпляра	10	041	
20 27	/EC4_BASE/BOOK/			0		. Q	041	a n
⊿ / ລວ	VECT DAGE VOON /	TUT	TUT	0	типу литературы	10	J J Z T C I I	a
∠0	/ TOOUL /	1/20	1/20	0	номеру ксу	ΤU	130	

продолжение файла ELECAT.DBF "в ширину"

	TAG2	SF2	TAG3	SF3	TAG4	SF4	TAG5	SF5	DICT	REPL
	"C"	"C"	"C"	"C"	"C"	"C"	"C"	"C"	" C "	"N"
1										0
2										0
3										0
4										0
5										0
6										0
7										0
8										0
9										0
10										0
11	700	а								0
12										0
13	710	а	111	а	711	а	245	С		0
14										0
15	650x									0
16								$\langle E($	C4_SUPP\DIC\KEYALL.DBF	0
17								E(	C4_SUPP\DIC\D_092A.DBF	1
18										0
19										0
20										0
21										0
22										0
23										0
24										0
25										0
26										0
27								$\langle E($	C4_SUPP\DIC\D_952A.DBF	0
28										0

Комментарии:

NB! Данные в колонке NTX должны всегда совпадать с данными в колонке FILE\_TYPE

Записи с номерами 1,3,5,6,9,10 в текущей версии не используются.

С 1 по 10 записи располагаются описания файлов системной части.

Во второй записи содержится имя файла перекрестных ссылок (МЕМО). Оно строится из данных колонок РАТН (путь), FILE\_NAME (имя файла), TYPE (расширение). Если в файле конфигурации (BOOK.CNF) задана подстановка (например, DSK N), то полное имя файла перекрестных ссылок будет: N:\EC4\_BASE\BOOK\MEMO.DBF

NB! Файлы MC\*.DBF, MC\*.NTX будут располагаться в той же директории, что и файл MEMO.DBF

NB! Имя файла перекрестных ссылок должно всегда быть МЕМО

В четвертой записи содержится имя файла описания полей. Оно строится из данных колонок РАТН (путь), FILE\_NAME (имя файла), TYPE (расширение). Если в файле конфигурации (BOOK.CNF) задана подстановка (например, DSK N), то полное имя файла описания полей будет: N:\EC4\_CNFW\BOOK\FIELD.DBF NB! Файлы FIELD.DBT, FIEI1.NTX, FIEI2.NTX должны располагаться в той же директории, что и файл FIELD.DBF

NB! Имя файла описания полей должно всегда быть FIELD

В седьмой записи содержится имя файла паролей. Оно строится из данных колонок PATH (путь), FILE\_NAME (имя файла), TYPE (расширение). Если в файле конфигурации (BOOK.CNF) задана подстановка (например, DSK N), то полное имя файла паролей будет: N:\EC4\_CNFW\BOOK\USER.DBF

NB! Имя файла описания полей должно всегда быть USER

В восьмой записи содержится путь к директории словарей. Он строится из данных колонки РАТН (путь). Если в файле конфигурации (BOOK.CNF) задана подстановка (например, DSK N), то полный путь к директории словарей будет: N:\EC4\_SUPP\DIC

NB! Не ищите во всем сказаном выше смысла! Его нет. Просто, так должно быть – и все! Если хотите, то так сложилось исторически. Когда-нибудь авторы программы намереваются навести со всем этим порядок. Когда-нибудь...

Начиная с 11-й записи расположены описания поисковых таблиц.

Имя файла поисковой таблицы складывается из данных в колонках РАТН (путь), FILE\_NAME (имя файла), а расширение у него всегда DBF. Если в файле конфигурации (BOOK.CNF) задана подстановка (например, DSK N), то полное имя файла поисковой таблицы будет: N:\EC4\_BASE\BOOK\AUT.DBF

NB! Файлы AUT.NTX и AUT\_CN.NTX располагаются в той же директории, что и файл AUT.DBF

NB! Имя файла поисковой таблицы должно состоять из трех латинских букв в верхнем регистре и быть уникальным.

В колонке TITLE размещается русское название поисковой таблицы.

В колонке CODE стоит длина данных в поисковой таблице. При занесении информации данные перетранслируются в верхний регистр и либо дополняются до указанной длины пробелами, либо усекаются справа.

В колонке ТҮРЕ (записи с номером больше 10) заносится глубина индексирования. Так, значение 3 предполагает сортировку по первым трем символам.

NB! Значение колонки ТҮРЕ, равное 0, указывает на то, что индексирование проводится по всей длине данных.

В колонках TAG1, SF1, ... TAG5, SF5 указываются метки и коды подполей тех элементов библиографического описания, которые заносятся в поисковую таблицу. Так, для создания таблицы поиска по автору в приведенном примере указаны 100а и 700а – метки и коды подполей для первого автора и других авторов.

Вместо меток здесь может стоять имя файла, например, TIN, это означает, что в файле TIN.BLC (такое расширение обязательно, и начало имени файла с буквы Т тоже обязательно) описано, по каким правилам обрабатывать данные, прежде чем занести их в поисковую таблицу. Сам файл располагается в директории \EC402\EC4\_SUPP\BLC.

В колонке DICT указывается имя "внешнего словаря" системы, который подключается для выбора данных при поиске информации.

NB! Подстановка DSK N не действует при определении имени файла словаря.

В колонке REPL указывается тип "внешнего" словаря, это может быть 1 или 2.

#### FIELD.DBF - описание полей электронного каталога

В файле FIELD.DBF содержится описание полей электронного каталога. Структура файла FIELD.DBF достаточно сложная. Вносить изменения "вручную" не рекомендуется. Для этого лучше всего использовать меню "НАСТРОЙКА.БИБЛИОГРАФИЧЕСКИЕ ЭЛЕМЕНТЫ" (F9 из окна "Каталог", позиция Запись).

Для файла FIELD.DBF создаются два индексных файла. Первый файл – FIEI1.NTX, индекс – TAG+FIELD. Этот индекс необходим для быстрого поиска по метке. Второй файл – FIEI2.NTX, индекс – 1000\*ORDTAG+ORDFLD, индекс нужен для показа элементов в заданном порядке на экране.

## USER.DBF - файл паролей

В файле паролей содержится информация о пользователях, имеющих доступ к электронному каталогу.

NB! Внести в него изменения из программы невозможно.

Отредактировать файл можно с помощью любого редактора DBF файлов (например, DBU.EXE). Структура файла USER.DBF:

Тип поля	Длина поля
Character	20
Character	20
Character	32
Character	20
	Тип поля Character Character Character Character

После установки программы этот файл выглядит так:

NAME	PAROL	FIO	BILER
	1958		

Как видно, занесен один единственный пароль безымянного пользователя.

Имеют значения только первые две колонки: NAME и PAROL. При первом входе в редактор, либо при копировании записей, система интересуется паролем. Ищет она его во второй колонке файла USER.DBF. Если находит, то разрешает действия, если нет - то дальнейшие действия Вам запрещены.

Значение из первой колонки используется так: первые четыре символа заносятся в поле 001 каждой записи в электронном каталоге для идентификации человека, который занес запись, и последнего, кто ее редактировал. Больше они нигде пока не используются, но в дальнейшем использоваться будут.

#### LEVEL.CNF - файл библиографических уровней

Файл библиографических уровней задает список уровней для данного электронного каталога. Типичный пример приведен ниже:

- а: Глава (часть, раздел,...) книги, диссертации, ...
- b: Статья из периодического издания
- с: Коллекция, библиотека, собрание, ...
- d: Часть коллекции, библиотеки, собрания, ...
- m: Монография ( книга, диссертация, ... )
- s: Периодическое издание.

Именно эти элементы будут в меню при нажатии клавиши F3 из окна Основного меню системы при занесении новой записи. В первой колонке указан библиографический уровень, согласно формату USMARC (здесь представлены все возможные уровни, больше быть не может, а меньше пожалуйста).

NB! Не вносите изменения в первую колонку. Данные из нее заносятся в саму запись.

Во второй колонке - комментарий, Вы можете его изменить, отредактировав файл LEVEL.CNF.

Каждому библиографическому уровню соответствует свой список полей в файле FIELD.DBF.

Из программы можно установить умолчание на один из уровней, тогда при нажатии клавиши F3 курсор будет стоять именно на этом элементе. Умолчание устанавливается для каждого каталога отдельно.

Для установки умолчания из программы нажмите клавишу F3 из Основного меню системы (кроме окна Поиск по), переместите курсор на нужный элемент, нажмите клавишу F9; система выдаст сообщение, что умолчание установлено, для продолжения нажмите любую клавишу.

В файле конфигурации каталога за умолчание отвечает параметр BBL.

## КЕҮ.CNF - файл "внешних" функций

В системе предусмотрена возможность запрограммировать клавиши. То есть, "посадить" на клавиши выполнение "внешних" (по отношению к программе) функций. В файле KEY.CNF каждого каталога находится список клавиш и функций. Содержание файла Вы можете изменить. Пример такого файла приводится ниже:

```
291 {|| 244 }
-36 {|| ExeFileBlock( "TextQuir.blc" ) }
-19 {|| if(SetCW()==2.and.Set_Help()=="MAIN",ExeFileBlock( "ISBNChek.blc" ),
)
-34 {|| ExeFileBlock( "WriteBuf.blc" ) }
-31 {|| ExeFileBlock( "SetLink.blc" ) }
-32 {|| ExeFileBlock( "ShowAll.blc" ) }
-33 {|| ExeFileBlock( "ShowLink.blc" ) }
-37 {|| ExeFileBlock( "SetAll.blc" ) }
```

В первой колонке (первое слово в каждой строке) приводится код клавиши. Затем, через пробел, следует блок кода, который выполняется при нажатии соответствующей клавиши. Как правило, этот блок кода просто вызывает выполнение другого файла с блоками кода (функция ExeFileBlock(), а ее параметр и есть имя другого файла с блоками кодов). Так, например, в строке:

-34 { || ExeFileBlock( "WriteBuf.blc" ) }

указано, что при нажатии клавиш Alt+F5 (код -34) будут выполнены действия, записанные в файле описания с именем WRITEBUF.BLC.

Блок кода может возвращать число. В этом случае это число будет интерпретироваться как результат действия клавиши с таким кодом. Так, в строке:

291 { | | 244 }

происходит просто перекодировка кода 291 (бывают такие компьютеры Российского производства, которые упорно посылают этот код при нажатии клавиши со строчной русской буквой р) в код 244.

NB! Таблицу кодов функциональных клавиш, принятую в Clipper, смотрите в Приложениях. Кодировка остальных клавиш совпадает с общепринятой.

Описание функций, которые могут быть использованы, смотрите в главе Файлы описания форматов экрана, документов, клавиш и файлы перекодировки.

NB! Ряд клавиш (F3, F4 и др.) уже задействованы в программе, поэтому на эти клавиши лучше "не покушаться".

Если в файле конфигурации каталога, например, BOOK.CNF, указана настройка пути к файлам блока кода для "внешних" функций, например, PKB C:\EC402\EC4\_SUPP\KEY, то поиск файлов, указанных как параметр функции ExeFileBlock(), вначале будет производиться в текущей директории, а затем в директории, приведенной после описателя PKB (например, в C:\EC402\EC4\_SUPP\KEY). DOCUMENT.CNF - список файлов описания документов

В файле DOCUMENT.CNF содержится список файлов описания документов для данного электронного каталога. При вызове документа из программы происходит подстановка диска (если указано, например, DSK N в файле BOOK.CNF) в имя файла описания документа, а затем его выполнение.

Вид файла DOCUMENT.CNF для каталога Книги:

С:\EC402\EC4\_SUPP\DOC\L001\_42.DOC Список литературы C:\EC402\EC4\_SUPP\DOC\B01g\_42.DOC Бюллетень C:\EC402\EC4\_SUPP\DOC\I001\_42.DOC Инвентарная книга C:\EC402\EC4\_SUPP\DOC\I002\_42.DOC Лист книги суммарного учета \_\_\_\_\_\_Имена файлов описания документов Названия, которые доступны пользователю при работе с

программой в текущем каталоге.

CONVERT.CNF - список файлов перекодировки (конверторов)

В файле CONVERT.CNF содержится список файлов перекодировки (конверторов). С помощью файлов перекодировки задаются:

- 1. Групповые преобразования в электронном каталоге. Например, можно изменить те или иные данные во всем каталоге или в выбранном подмножестве записей.
- 2. Преобразование информации при загрузке данных.
- 3. Преобразование информации при выгрузке данных.

При вызове файла перекодировки из программы происходит подстановка диска (если указано, например, DSK N в файле BOOK.CNF) в имя файла перекодировки, а затем его выполнение.

Пример файла CONVERT.CNF для каталога Книги:

C:\EC402\EC4\_SUPP\CNV\NEW.CON Из 2-й в 3-ю или 4-ю по базе C:\EC402\EC4\_SUPP\CNV\NEWLOAD.CON Из 2-й в 3-ю или 4-ю загрузка C:\EC402\EC4\_SUPP\CNV\LDB\_LD1.CON Пример загрузки из LDB файлов C:\EC402\EC4\_SUPP\CNV\GPNTB.CON Вывод для ГПНТБ

Формат файла такой же, что и DOCUMENT.CNF

ADDCARD.CNF - список возможных добавочных описаний

В файле ADDCARD.CNF содержится список библиографических элементов на которые могут создаваться добавочные карточки. Например:

100а Первый автор 700а Другие авторы 110а Пост. кол. автор 111а Врем. кол. автор ...

В первой колонке указана метка библиографического элемента, во второй колонке его название, которое может быть произвольным. Настройку файла можно производить из программы. Смотрите раздел Добавочная карточка.

Настройка каталогов Отключение каталога Подключение уже существующего каталога Изменение имени каталога Создание нового каталога Изменение параметров каталога
Настройка поисковых таблиц Общие замечания о построении поисковых таблиц Создание новой поисковой таблицы Изменение параметров поисковой таблицы Реактуализация
Настройка элементов ввода Общие замечания Создание нового элемента ввода данных Редактирование параметров элемента ввода Дополнительные возможности
Настройка файлов описания документов и файлов перекодировки Отключение файла описания документа Подключение уже существующего файла описания документа Изменение параметров файла описания документа

Настройка проверки на дублетность

#### Настройка каталогов

#### Отключение каталога

NB! Производить настройку системы имеет право пользователь, имеющий права на чтение/запись и полномочия администратора системы.

У пользователя всегда есть возможность отключить каталог.

NB! Нельзя отключить каталог, который является текущим. Вы не можете отключить, например, каталог Подписка, находясь в нем.

Если все-таки будет сделана попытка отключить текущй каталог, то система выдаст предупреждающее сообщение о том, что это невозможно. Для продолжения работы нажмите Enter.

Существуют два варианта отключения каталога. Если Вы находитесь в каталоге, который по списку расположен выше отключаемого каталога (например, в каталоге Книги), то все пройдет без дополнительных трудностей. Но если Вы находитесь в каталоге, расположенном ниже по списку, чем отключаемый каталог, то система выдаст сообщение:

> При отключении этого каталога Вы выйдете из системы > Да < Нет

Это означает, что при выборе элемента Да система отключит каталог и Вы выйдете из программы. После этого Вам придется снова заходить в систему. Такого варианта отключения всегда можно избежать, если только отключаемый каталог не является первым в списке.

В качестве примера отключим каталог Подписка. При этом текущим будет каталог, например, Комплектование.

- 1. Перейдите в каталог Комплектование.
- Войдите в меню Список каталогов. Для этого переместите курсор в позицию Каталог в окне Каталог. Нажмите клавищу F9.
- Отключите каталог Подписка. Переместите курсор на элемент Подписка. Нажмите клавишу F6. Система запросит подтверждение на отключение каталога:

Отключить	каталог ?	
> Да <	Нет	

NB! Если бы Вы находились в каталоге, например, Читатели (по списку он расположен ниже каталога Подписка), то система выдала бы сообщение о том, что при отключении Вы выйдете из программы.

При выборе элемента Да система выдаст в верхней строке экрана:

Отключение каталога ...

После этого каталог будет отключен.

4. Возврат в Основное меню системы. Нажмите клавишу Esc для выхода из меню Список каталогов.

Подключение уже существующего каталога

В этом разделе будет рассмотрен пример подключения уже существующего каталога. Создание нового каталога и его подключение изложено в следующих разделах этой главы.

Подключим каталог Подписка, который был отключен в предыдущем примере.

При подключении каталога не имеет значения какой каталог является текущим.

- 1. Войдите в Меню каталогов. Для этого нажмите клавишу F9, курсор в позиции Каталог в окне Каталог Основного меню системы.
- 2. Подключите каталог. Нажмите клавишу F3. На экране меню:

— ПОДКЛЮЧЕНИЕ КАТАЛОГА—

```
Файл: \EC402\EC4_CNFW\.CNF
Имя:
```

Помощь<F1> Coxp.<F4> Выход<Esc>

В поле Файл введите имя файла описания каталога, который Вы хотите подключить. В нашем случае введите:

\EC402\EC4\_CNFW\PODP\PODP.CNF

Нажмите Enter для перехода на элемент Имя, введите название каталога. Под этим именем каталог будет в списке подключенных каталогов. Введите, например:

Подписка на 1996 (июль-август)

Можно задать любое другое имя каталога.

Далее приведены имена файлов описания каталогов после установки системы в директорию EC402.

Файлы конфигурации для пользователя с правами только на чтение:

Название каталога	Имя директории	Имя файла описания
Книги	\EC402\EC4_CNFR\BOOK	BOOK.CNF
Статьи	\EC402\EC4_CNFR\BIBL	BIBL.CNF
Периодика	\EC402\EC4_CNFR\PERI	PERI.CNF
Книги_МГУ	\EC402\EC4_CNFR\LMSU	LMSU.CNF

Файлы конфигурации для пользователя с правами на чтение/запись:

Название каталога	Имя директории	Имя файла описания
Книги	\EC402\EC4_CNFW\BOOK	BOOK.CNF
Статьи	$EC402 EC4_CNFW BIBL$	BIBL.CNF
Комплектование	$EC402 EC4_CNFW COMP$	COMP.CNF
Периодика	\EC402\EC4_CNFW\PERI	PERI.CNF
Подписка	$EC402 EC4_CNFW PODP$	PODP.CNF
Читатели	$EC402 EC4_CNFW READ$	READ.CNF
Книги_МГУ	$EC402 EC4_CNFW LMSU$	LMSU.CNF

- Сохраните заданные параметры. Для подключения каталога достаточно нажать клавишу F4. Вы вернетесь в меню Список каталогов, и последним будет каталог Подписка на 1996 (июль-август).
  - NB! Если имя файла описания задано не верно, то система выдаст сообщение о том, что имя файла задано не верно. Для продолжения работы нажмите Enter.
  - NB! Если не будет задано имя каталога, то система подключит каталог без имени.
- 4. Возврат в Основное меню системы. Нажмите клавишу Esc.

## Изменение имени каталога

Изменение имени каталога по сути является одним из элементов изменение параметров каталога, об этом подробно рассказано далее. Здесь, не углубляясь в детали, мы расскажем, как изменить имя каталога.

Например, изменим имя каталога Статьи на имя Статьи из журналов.

- Перейдите в режим Меню каталогов. Для этого нажмите клавишу F9 из окна Каталог Основного меню системы, курсор в позиции Каталог.
- Перейдите в меню Редактирование каталога. Переместите курсор на элемент Статьи и нажмите клавишу F4. На экране довольно сложное меню. Вам надо перейти на элемент Имя каталога (клавиша стрелка-вниз), в этом поле ввести:

Статьи из журналов

Для сохранения нового имени достаточно нажать клавишу F4.

Ваш каталог в списке каталогов появляется под новым именем.

3. Возврат в Основное меню системы. Нажмите клавишу Esc.

Создание нового каталога

При инсталляции системы происходит установка стандартных каталогов. У пользователя есть возможность создавать новые каталоги.

При создании нового каталога в системе реально происходит копирование конфигурационных файлов одного из уже подключенных каталогов. Почему именно так? Потому что с каждым каталогом связано много файлов

конфигурации, и с каждой новой версией количество их будет расти. Для того, чтобы не возникло трудностей с определением файлов для нового каталога, выбирается из уже имеющихся каталогов наиболее подходящий и копируются все его файлы конфигурации в новую директорию. Далее можно будет произвести донастройку созданного каталога.

В качестве примера создадим новый каталог с именем BNEW и русским названием "Книги (новые)".

- 1. Войдите в Меню каталогов. Для этого нажмите клавишу F9, курсор в позиции Каталог в окне Каталог Основного меню системы.
- Скопируем файлы конфигурации наиболее подходящего каталога. В нашем примере надо скопировать параметры каталога Книги. Переместите курсор на элемент Книги и нажмите клавишу F5. На экране меню:

КОПИРОВАНИЕ КАТАЛОГА Каталог: Книги Имя (4 лат.буквы): Название: Файл описания каталога: C:\EC402\EC4\_CNFW\????\ELECAT.DBF Файл базы: C:\EC402\EC4\_BASE\????\MEMO.DBF Файл библ.элементов: C:\EC402\EC4\_CNFW\????\FIELD.DBF Файл паролей: C:\EC402\EC4\_CNFW\????\USER.DBF Доступ: (\*) Локальный Директория словаря: C:\EC402\EC4\_SUPP\DIC\ () Удаленный Директория поиска: C:\EC402\EC4\_BASE\???\

Для перемещения по пунктам и элементам используйте клавиши: стрелка-вверх, стрелка-вниз, Enter, Tab, Shift+Tab.

Помощь<F1> Coxp.<F4> Выход<Esc>

В поле Каталог высвечивается имя каталога, с которого будет сделана копия, перейти курсором в это поле невозможно.

В поле Имя введите уникальное имя каталога (4 латинских буквы в верхнем регистре).

NB! Имена каталогов должны быть уникальными и никогда не повторяться!

Нажмите клавишу Enter для перехода в следующее поле, возможно, курсор сам перейдет на следущее поле. В этот момент произойдет фактическое копирование файлов конфигурации и создание необходимой структуры директорий.

NB! Как только Вы перешли на поле Название новый каталог уже создан. Даже если потом для выхода из меню Вы воспользуетесь клавишей Esc, то он будет существовать, но будет не подключенным к системе. Вы в последствии можете его подключить.

На экране появятся изменения:

---КОПИРОВАНИЕ КАТАЛОГА--

Каталог: Книги

Имя (4 лат.буквы): BNEW Название:

```
Файл описания каталога: \EC402\EC4_CNFW\BNEW\ELECAT.DBF
Файл базы: \EC402\EC4_BASE\BNEW\MEMO.DBF
Файл библ.элементов: \EC402\EC4_CNFW\BNEW\FIELD.
Файл паролей: \EC402\EC4_CNFW\BNEW\USER.DBF Доступ:
(*) Локальный
Директория словаря: \EC402\EC4_SUPP\DIC\ () Удаленный
Директория поиска: \EC402\EC4_BASE\BNEW\
```

Помощь<F1> Coxp.<F4> Выход<Esc>

По сути дела, все. Остается только внести в поле Название русское имя каталога, под которым в дальнейшем он будет фигурировать в списке каталогов. В нашем случае это название "Книги (новые)" но при желании это может быть любая комбинация символов и цифр.

Для подключения нового каталога нажмите клавишу F4. При правильном вводе данных новый каталог будет подключен к системе. Настройка списка элементов редактирования и поисковых таблиц в новом каталоге осуществляется по обычным правилам после перехода в этот каталог.

NB! Вновь созданный каталог всегда будет последним в списке. Переместить каталог в списке возможно, для этого надо выйти из программы и поменять местами строчки в файле EC.CNF в соответствии с Вашими желаниями (файл расположен в той же директории, что и файл EC4.EXE). Затем войдите снова в систему.

Что же произошло в действительности?

1. Была создана директория для размещения информации электронного каталога. В примере:

 $EC402 EC4_BASE BNEW$ 

 Была создана директория в директории конфигурации электронных каталогов:

\EC402\EC4\_CNFW\BNEW

- 3. В директорию \EC402\EC4\_CNFW\BNEW были скопированы все файлы конфигурации из директории \EC402\EC4\_CNFW\BOOK.
- 4. Имя файла конфигурации \EC402\EC4\_CNFW\BOOK\BOOK.CNF было заменено на имя \EC402\EC4\_CNFW\BNEW\BNEW.CNF.
- 5. В файлах BNEW.CNF, ELECAT.DBF название директории \BOOK поменялось на название директории \BNEW.

Для дальнейшей настройки нового каталога необходимо перейти в этот каталог и использовать стандартные меню для настройки поисковых таблиц, списка полей, списка документов и т.д.

Изменение параметров каталога

В предыдущих примерах было рассказано о возможности изменения имени каталога, но в системе также предусмотрено изменение имени файла конфигурации каталога, путей к файлам описания элементов редактирования, паролей, файлу конфигурации; дается возможность изменить имя директории, где размещаются словари.

NB! Нельзя с помощью средств системы Библиотека 4.0 изменить место расположения файлов с записями данного каталога, а также поисковых таблиц. Это можно сделать только с помощью средств операционной системы, но затем надо перенастроить путь к этим файлам в файле ELECAT.DBF, что должно быть сделано в данном случае также с помощью средств DOS.

Для примера рассмотрим изменение стандартного имени каталога Статьи на имя Статьи (законы), а также изменение имени файла конфигурации с \EC4\BIBL\BIBL на \EC4\BIBL\ART\_LOW.

1. Войдите в режим Меню каталогов. Для этого из Основного меню системы, курсор в позиции Каталог, нажмите клавишу F9. Переместите курсор на элемент Статьи.

2. Измените параметры каталога. Нажмите клавишу F4, и Вы попадете в меню Редактирование каталога.

РЕДАКТИРОВАНИЕ КАТАЛОГА Файл конфигурации: \EC4\_CNFW\BIBL\BIBL.CNF Имя каталога: Cтатьи Файл описания каталога: \EC4\_CNFW\BIBL\ELECAT.DBF Файл библ.элементов: \EC4\_CNFW\BIBL\FIELD.DBF Файл паролей: \EC4\_CNFW\BIBL\USER.DBF Доступ: (.) Локальный Директория словаря: \EC4\_SUPP\DIC\ () Удаленный Помощь<F1> Coxp.<F4> Выход<Esc>

Измените данные, например, так:

— РЕДАКТИРОВАНИЕ КАТАЛОГА-

Файл конфигурации: \EC4\_CNFW\BIBL\ART\_LOW.CNF Имя каталога: Статьи (законы) Файл описания каталога: \EC4\_CNFW\BIBL\ELECAT.DBF Файл библ.элементов: \EC4\_CNFW\BIBL\FIELD.DBF Файл паролей: \EC4\_CNFW\BIBL\USER.DBF Доступ: (.) Локальный Директория словаря: \EC4\_SUPP\DIC\ () Удаленный Помощь<F1> Coxp.<F4> Выход<Esc>

NB! При изменении имени директории необходимо, чтобы она уже существовала.

Для сохранения новых параметров каталога нажмите клавишу F4. Система будет запрашивать подтверждение на изменение всех имен файлов. Выбирайте Да для подтверждения изменения имени файла, при выборе Нет Вы вернетесь в меню Редактирование каталога. Для выхода без изменений параметров каталога нажмите клавишу Esc. Общие замечания о построении поисковых таблиц

Поисковые таблицы системы позволяют проводить быстрый поиск информации. Они автоматически обновляются при вводе новых записей, редактировании и удалении записей.

Список поисковых элементов задается произвольно в любой момент работы системы. Однако если Вы уже накопили информацию и только после этого решили создать новую поисковую таблицу, Вам придется занести в эту таблицу ранее введенную информацию. Этот процесс называется реактуализацией.

NB! Поэтому еще до начала ввода информации продумайте, по каким признакам Вы хотите ее искать, создайте соответствующий перечень поисковых элементов, и только после этого приступайте к вводу информации.

В любые поисковые таблицы информация по умолчанию заносится заглавными буквами (это можно изменить). В поисковую таблицу возможно занесение до пяти меток USMARC. Например, Вы создаете таблицу для поиска по авторам. Вы можете туда занести как первого автора (метка 100а), так и всех остальных авторов (метка 700а), а также постоянных коллективных авторов (метка 110а) и т.д. А можете построить отдельную поисковую таблицу для коллективных авторов.

Есть возможность и более сложной обработки информации перед занесением в поисковую таблицу. Можно сделать свертку из нескольких полей, можно обрабатывать заглавие книги и выделять из него ключевые слова, возможно разбиение ключевых слов по пробелам и занесение в таблицу как словосочетания целиком, так и отдельных его частей, и многое другое, на сколько хватит Вашей фантазии. Для этого надо указывать не метку USMARC, а трехзначный код, который должен начинаться с символа t. Например, t14.

Но прежде чем такой код указывать (где его указывать, Вы узнаеие, прочитав следующий раздел), Вы должны создать файл t14.blc (расширение blc обязательно), в нем как раз и должны быть описаны правила занесения информации в поисковую таблицу. Это один из файлов блока кода, принципы его построения аналогичны файлам описания форматов экрана и документа. Последним оператором там должна быть функция Send( exp ). В качестве параметра, передаваемого в нее, может быть строка или массив строк, которые должны быть занесены в поисковую таблицу.

Файл с таким же именем, но расширением get, может быть создан для ввода текста запроса и дальнейшей его обработки. Например, если в качестве запроса вам надо ввести Год издания и Название издательства, то лучше это сделать с помощью такого меню:

> Год издания: Издательство:

вызов которого Вам и надо написать в файле g14.get.

Переход в режим Меню поисковых элементов, где можно производить настройку поисковых элементов, происходит из окна Поиск по Основного меню системы при нажатии клавиши F9. На экране меню:

	МЕНЮ ПОИСКОВЫХ	К ЭЛЕМЕНТОВ
Cos	здать <f3> Удализ</f3>	гь <f8> Bce<grey+></grey+></f8>
	Редакт. <f4></f4>	Реактуал. <f6></f6>
	автору	AUT
	заглавию	TIT
	коллек.автору	COR
	названию серии	SER
	предмет.рубрика	TER
	ключевым словам	KEY
	индексам ГАСНТИ	GAS
<b></b>	Помощь <f1></f1>	Выход <esc></esc>

Находясь в этом режиме, Вы можете создать новую поисковую таблицу, изменить параметры старой таблицы, удалить таблицу, а также сделать реактуализацию одной, нескольких или всех поисковых таблиц.

Для перемещения по элементам пользуйтесь клавишами: стрелка-вверх, стрелка-вниз, Ctrl+PgUp, Ctrl+PgDn. Для выбора (отмены выбора) элемента служит клавиша Пробел, для выбора всех элементов воспользуйтесь клавишей Серый плюс. Для отмены выбора всех элементов служит клавиша Серый минус.

Создание новой поисковой таблицы

Создадим поисковую таблицу для элемента Каталожные индексы. При обработке изданий в это поле в редакторе записи в библиотеке Московского университета заносятся наши собственные систематические индексы. Метка USMARC - 090с.

- Войдите в режим Меню поисковых элементов. Для этого из окна Поиск по Основного меню системы нажмите клавищу F9.
- Прейдите в меню Поисковый элемент.Установка параметров в режиме создания нового поискового элемента. Для этого нажмите клавишу F3. На экране меню:

--ПОИСКОВЫЙ ЭЛЕМЕНТ.УСТАНОВКА ПАРАМЕТРОВ-

Файл: C:\EC402\EC4\_BASE\BOOK\.DBF

Метки: Первая: Вторая: Третья: Четвертая: Пятая: Поиск по: Индекс: Длина:

Словарь: Тип словаря: Это один из типов меню Установка параметров. Каждая поисковая таблица характеризуется следующими параметрами:

- Файл Имя файла (вместе с путем), где будет размещаться поисковая таблица. Расширение должно быть обязательно DBF. Длина имени обязательно равна трем символам, и первые три символа имени не должны совпадать ни с одной другой поисковой таблицей данного каталога. Это связано с тем, что при поиске с помощью языка запросов идентификатор элемента состоит из трех символов и берутся первые три символа от имени файла поисковой таблицы.
  - NB! Имена файлов поисковых таблиц в Библиотеке 4.02 должны состоять из трех символов латинского алфавита!

Находясь на этом элементе, при нажатии клавиши Пробел, Вы перейдете в режим Выбор файла.

- Первая В поле заносится метка USMARC формата. Если в записи есть данные в этой метке, то они будут занесены в поисковую таблицу. В остальные элементы: Вторая, Третья, Четвертая, Пятая заносятся остальные метки. Полный список меток дан в Приложениях.
- Поиск по В поле указывается название признака поиска. В дальнейшем именно под этим именем он будет виден в окне Поиск по.
- Индекс Заданная в этом поле величина определяет длину индекса этой поисковой таблицы. Так, если будет задано значение 2, то быстрый поиск будет произведен по первым двум символам; заданное слово будет найдено, даже если у него длина 30 символов, но для этого потребуется больше времени, нежели если бы было задано значение 10. Также только по двум символам будет идти поиск во внутреннем словаре, и записи в словаре будут отсортированы только по первым двум символам.

При задании этого поля надо руководствоваться принципом разумности. Чем больше длина индекса, тем быстрее поиск, но тем больше поисковая таблица будет занимать места на диске.

- Длина Поле задает длину поискового элемента, т.е. сколько символов поля будет попадать в таблицу; длинные слова будут обрезаться до заданного значения. Чем больше длина заносимых слов, тем больше занимает таблица места на диске.
- Словарь Поле содержит имя внешнего файла словаря, который будет использоваться для выбора элементов поиска (клавиша F7 в окне Поиск по, если имя задано).

Тип словаря Определяет тип словаря, т.е. является ли внешний словарь просто списком (тип 2) или это код и его значение (тип 1).

В поле Файл введите, например:

C:\EC402\EC4\_BASE\BOOK\LIL.DBF

Нажмите клавищу Enter для перехода в поле Первая. Введите метку USMARC. В нашем примере это:

090c

В случае необходимости занесения нескольких меток в одну поисковую таблицу, надо заполнить и остальные элементы пункта Метки.

Если Вы хотите обрабатывать информацию по своим правилам, то вместо меток Вы должны указать, например, t14. Такое имя должно быть задано только одно.

Нажмите клавишу Tab, для перехода на поле Поиск по введите:

#### каталожным индексам

Нажмите клавишу Enter. В поле Индекс введите 0, это означает, что индекс будет построен по всей длине элемента, заданной в поле Длина. Нажмите клавишу Enter для перехода на это поле и введите:

10

Оставшиеся два поля можно оставить не заполненными, если у Вас нет внешнего словаря для этого поля, или заполнить позднее, когда словарь появится.

- Сохраните параметры нового поискового элемента. Для этого нажмите клавишу F4. Если параметры были заданы верно, то элемент будет создан и подключен к концу поисковой таблицы.
- 4. Возврат в Основное меню системы. Нажмите клавишу Esc.
- NB! Если у Вас уже есть наполненный ЭК и Вы создали новую поисковую таблицу, то обязательно проведите реактуализацию старых записей по новой поисковой таблице.
- NB! Порядок поисковых элементов может быть изменен.

Изменить порядок можно только из режима Меню поисковых элементов, клавиша F9 из окна Поиск по Основного меню системы. Для перемещения элемента перед текущим служит клавиша A (латинский шрифт), для перемещения после текущего – клавиша B (латинский шрифт). На экране появится меню:

```
ПЕРЕНОС ЭЛЕМЕНТА
Вставка между: заглавию.....
коллек.автору...
Описание:
```

Выбор<Enter> Выход<Esc>
В первых двух строках высвечиваются названия элементов, между которыми будет вставлен переносимый элемент. В поле Описание введите имя поисковой таблицы, под которым она высвечивается в окне Поиск по и которую Вы хотите переместить, нажмите клавишу Enter; порядок элементов будет изменен.

Любой элемент может быть удален. Для этого в режиме Меню поисковых элементов переместите курсор на нужный элемент и нажмите клавишу F8. Система запросит подтверждение на удаление элемента, при выборе Да из поисковой таблицы будет удален отмеченный Вами элемент, при выборе Нет Вы вернетесь в предыдущее меню без изменений.

#### Изменение параметров поисковой таблицы

При необходимости Вы можете изменить любые параметры поисковой таблицы, но после изменения некоторых из них надо проводить реактуализацию. Предположим, Вы ввели 1000 записей и после этого поняли, что в эту таблицу надо заносить также данные и из другого поля. Новые записи будут обрабатываться в соответствии с новыми установками, но старые записи уже обработаны по старым правилам. В этом случае для изменения уже занесенных элементов служит реактуализация.

Процесс реактуализации надо проводить, если Вы:

создали новую поисковую таблицу; изменили набор меток; увеличили длину поисковых элементов; изменили глубину индексирования.

Если длина поисковой таблицы была уменьшена, то система создаст новую таблицу с заданными параметрами.

В качестве примера рассмотрим изменение параметров поискового элемента для поиска по ключевым словам в каталоге Книги. После инсталляции в эту поисковую таблицу будут заноситься строки до 25 символов и индекс будет построен по всей длине. Изменим длину заносимых строк до 40, а индекс до 30.

- 1. Перейдите в окно Поиск по Основного меню системы.
- 2. Войдите в режим Поисковый элемент. Установка параметров в режиме редактирования.

Для этого переместите курсор на элемент ключевым словам и нажмите клавишу F4. На экране знакомое по предыдущему примеру меню, но оно уже заполнено информацией о текущем элементе.

- 3. Внесите изменения. В поле Индекс должно стоять значение 30, в поле Длина 40.
- 4. Сохраните изменения. Нажмите клавишу F4. Для выхода без изменений воспользуйтесь клавишей Esc. При сохранении с новыми параметрами система сообщит:

Изменение длины индекса ...

Далее будет выдано предупреждающее сообщение о том, что необходимо провести реактуализацию записей. Если Вас это не пугает, то выбирайте элемент Продолжать и нажимайте клавищу Enter.

5. Возврат в Основное меню системы. Нажмите клавищу Esc.

Если поисковая таблица содержит большое количество элементов, то создание таблицы с новыми параметрами может занять много времени. Изменение параметров сопровождается сообщением системы:

Изменение параметров словаря ...

Процесс будет закончен, как только исчезнет это сообщение.

# Реактуализация

Реактуализацию необходимо проводить после создания новой поисковой таблицы, при изменении перечня меток и при увеличении длины поискового элемента.

Реактуализировать можно как все поисковые таблицы сразу, так и любую возможную их комбинацию. Для выбора всех поисковых элементов служит клавиша Серый плюс в режиме Меню поисковых элементов. Клавиша Серый минус предназначена для отмены выбора всех элементов. Для выбора (или отмены выбора) одного элемента служит клавиша Пробел.

NB! Для ускорения процесса реактуализации и для дальнейшей корректной работы программы перед проведением реактуализации очистите поисковую таблицу. Для этого нажмите клавиши Shift+F8, стоя на соответствующей поисковой таблице.

Рассмотрим в качестве примера реактуализацию записей по вновь созданному поисковому элементу - каталожным индексам. Реактуализировать будем все записи электронного каталога.

- 1. Перейдите в окно Поиск по Основного меню системы.
- Войдите в режим Меню поисковых элементов. Для этого нажмите клавищу F9.
- 3. Отметьте элемент(ы) для реактуализации. В нашем случае надо отметить один элемент - каталожным индексам. Переместите курсор на этот элемент и нажмите клавишу Пробел. Напротив элемента появится символ []. Обнулите поисковую таблицу, нажав клавиши Shift+F8. В появившемся сообщении выберите элемент продолжать и нажмите клавишу Enter.
- Запустите процесс реактуализации. Для перехода в режим Реактуализация.Установка параметров нажмите клавишу F6. Установите диапазон записей для реактуализации. В нашем примере должны быть выбраны элементы Каталог и Все.

Для начала реактуализации нажмите клавишу F4. На экране

появится меню индикации процесса реактуализации. Для возврата в предыдущее меню без реактуализации нажмите клавищу Esc.

5. Возврат в Основное меню системы. Дождитесь окончания реактуализации (или прервите ее, нажав клавишу Esc). Для возврата в Основное меню нажмите клавишу Esc.

Настройка элементов ввода

#### Общие замечания

При вводе и редактировании информации в любом из каталогов Вы заполняете длинный список полей системы. В зависимости от каталога, а также от библиографического уровня, список меняется. В Приложениях даны списки всех полей для каждого из каталогов.

Вы можете помимо уже созданных полей добавлять свои собственные элементы, а также менять их параметры: название, длину, порядок. Можно менять метки и повторяемость, но здесь надо быть внимательным, т.к. хотелось бы, чтобы эти изменения проводились на основе стандарта USMARC.

Список полей ввода для каждого каталога хранится в файле базы данных. Этот файл, после инсталляции, расположен в директории конфигурации каждого каталога. Имя у него -FIELD.DBF. Так, для каталога Книги это будет файл \EC402\\EC4\_CNFW\BOOK\FIELD.DBF, для каталога Статьи -\EC402\EC4\_CNFW\BIBL\FIELD.DBF и т.д.

Полный список полей системы для текущего каталога вызывается по клавише F9 из окна Каталог, курсор в позиции Тек.записи:

	——H.	АСТРОИКА.	БИБЛИОГРА	ФИЧЕСКИЕ	ЭЛЕМЕНТЫ-	
	Cos	пать < F3>	Репак <	<b>F4&gt;</b> Vπa	пить < F8>	
	000,	daid (1.5)	гедак. ч	гт одо		
г-Порядо	ок——	-Метка	Назва	ние	Длина-	Уровень-
10 /	1	1008	Тип ФИО	0,1,2,3	1	abm
10 /	2	100a	Первый а	втор ФИО	48	abm
10 /	3	100b	Династ.н	омер	16	abm
10 /	4	100c	Титул (з	ван.)	24	abm
10 /	5	100d	Дата		16	abm
10 /	6	100e	Роль лиц		24	abm
20 /	1	7008	Тип ФИО	0,1,2,3	1	abm
20 /	2	700a	Другие а	вт.ФИО	48	abm
20 /	3	700e	Роль лиц		24	abm
30 /	1	1108	Порядок	0,1,	1	ms
30 /	2	110a	Пост.кол	.автор	180	ms
L						
		Помощ	ь <f1></f1>	Выход<Ез	SC>	

Меню состоит из пяти колонок: Порядок, Метка, Название, Длина, Уровень. Начнем с простых колонок. В колонке Метка содержатся метки элементов, они, как правило, соответствуют формату USMARC, если каталог содержит библиографическую информацию. В каталоге Читатели они взяты произвольно авторами программы, так что в этом каталоге Вы можете добавлять метку сообразно Вашим желаниям и не ориентироваться на формат USMARC. В колонке Название содержится название метки, под которым она выходит при вводе или редактировании информации. В колонке Длина задана длина поля. Но сейчас это уже не является ограничением, т.к. в редакторе записи есть клавиша F3, нажав которую Вы можете вводить бесконечно длинные тексты. В колонке Уровень указан библиографический уровень. Так, если стоит - ms, то это означает, что при вводе новой записи и выборе уровня m или s эта метка будет в списке элементов для заполнения, при выборе любого другого уровня этой метки на экране не будет.

Теперь остановимся на самой сложной колонке - Порядок. Эта колонка отвечает за порядок показа элементов на экране. Вы всегда можете изменить порядок элементов.

Первые три числа перед косой чертой отвечают за порядок поля. Так, если у Вас есть одно поле, например, 100, то первые три числа для этого поля должны совпадать для всех подполей этого поля. Если Вы измените 20 / 3 на 8 / 3, то элемент Роль лиц будет стоять первым в списке, но не забывайте, что порядок должен быть изменен и в двух оставшихся подполях этого поля: 20 / 1 на 8 / 1, 20 / 2 на 8 / 2

Числа за косой чертой отвечают за порядок подполей внутри поля. Так, если Вы измените 20 / 2 на 20 / 4 , то элемент Другие авт.ФИО у Вас будет расположен после элемента Роль лиц.

### Создание нового элемента ввода данных

Например, создадим новый элемент ввода данных с меткой 210а, согласно USMARC в это поле заносится сокращенное заглавие, поле не повторяемое.

- Перейдите в режим Настройка.Библиографические элементы. Для этого нажмите клавишу F9 из окна Каталог, курсор в позиции Тек.запись.
- Войдите в режим создания нового элемента. Нажмите клавишу F3. На экране меню:

	ЭПЕМЕНТА
	JIEHEIIIA
Название: Метка:	Уровень: Длина:
Повтор: Порядок	Имя словаря:
() Поля Предыдущий: 10 / 1	Тип словаря: О
() Подполя Текущий: /	
Следующий: 10 / 3	
Помощь <f1> Сохр.<f4></f4></f1>	Выход <esc></esc>

Для создания нового элемента надо заполнить следующие поля:

Название Содержит название элемента. Соответствует колонке Название в предыдущем разделе.

- Метка Четырехсимвольный код USMARC. Соответствует колонке Метка в предыдущем разделе.
- Уровень Содержит признак включения элемента в описание разного рода изданий; комбинацией всех возможных библиографических уровней является следующее сочетание: abcdms.
- Длина Цифровое значение, которое определяет длину данных поля.
- Повтор Задает повторяемость элемента.
- Поля Если этот элемент выбран, то будет повторяться все поле целиком со всеми входящими в него подполями.
- Подполя При выборе этого элемента повторяться будет только подполе в рамках поля. Выбран может быть только один элемент пункта Повтор. Возможно, что одни подполя поля повторяются внутри поля (подполе 852р - инвентарный номер), а другие подполя повторяются только вместе со всеми элементами поля (поле 700а -Другие авторы повторяется со всеми элементами этого поля). Если ни один из элементов пункта Повтор не выбран, то подполе не повторяемое.
- NB! Коды полей и подполей, а также повторяемость элементов необходимо согласовывать с описанием формата USMARC.
- Порядок Определяет месторасположение элемента на экране ввода.
- Предыдущий Показывает порядок предыдущего элемента (элемент, находящийся перед курсором в меню Настройка.Библиографические элементы). Элемент не доступен для ввода информации.
- Текущий Поле содержит порядок текущего элемента. При создании нового элемента в поле надо задать желаемый порядок, при редактировании элемента поле содержит установленный порядок элемента, который может быть изменен.
- Следующий Показывает порядок следующего элемента (элемент, находящийся после курсора в меню Настройка.Библиографические элементы). Элемент не доступен для ввода информации.
- Имя словаря Поле содержит имя внешнего словаря, который будет использоваться для ввода данных. Заполнение поля не обязательно.
- Тип словаря В поле задается тип внешнего словаря, если заполнено предыдущее поле. Может содержать одно из значений: 1 или 2. Подробнее о словарях см. главу Словари.

Для перемещения по элементам используйте клавиши:

стрелка-вверх, стрелка-вниз, PgUp, PgDn, Tab, Shift+Tab, Enter. Для выхода без создания нового элемента воспользуйтесь клавишей Esc.

В нашем примере заполните поля следующим образом:

ПАРАМЕТРЫ БИБПИОГРАФИЦЕСКОГО ЭПЕМЕНТА							
	J JIEMENIA						
Название: Сокращ.название	Уровень: т						
Метка: 210а	Длина: 100						
	Wwg openation						
повтор. порядок	имя словаря.						
() Поля Предыдущий: 10 / 1	Тип словаря: О						
( ) Подполя Текущий: 75 / 3							
Следующий: 10 / 3							
Помощь <f1> Сохр.<f4></f4></f1>	Выход <esc></esc>						

- NB! Если Вы создаете новый элемент для каталога Читатели, элементы которого нельзя отнести ни к одному библиографическому уровню, то задавать какой-либо уровень обязательно. Иначе этот элемент никогда не появится в списке элементов при создании новой записи или редактировании старой.
- Сохраните введенные значения. Для этого нажмите клавишу F4. Новый элемент будет включен в список. Место его расположения зависит от заданного Вами порядка. В данном случае он будет выходить на экран перед полями, описывающими серию.
- 4. Возврат в Основное меню системы. Нажмите клавишу Esc.
- NB! Общая рекомендация не следует задавать порядок элементов один за другим, т.к. в случае последующей вставки нового элемента Вам придется менять порядок у всех элементов, расположенных ниже нового. Оптимально задавать порядок полей с интервалом 10.

Редактирование параметров элемента ввода

У пользователя есть возможность отредактировать любой параметр элемента ввода данных. Например, изменим месторасположение элемента Инвент.номер так, чтобы он находился сразу после элемента Количество экз., и изменим длину 24 на 10.

- Перейдите в режим Настройка.Библиографические элементы. Для этого нажмите клавишу F9 из окна Каталог, курсор в позиции Тек.запись.
- Войдите в режим редактирования параметра элемента. Переместите курсор на нужный элемент, в данном случае на элемент Инвент.номер, и нажмите клавишу F4. На экране знакомое по предыдущему примеру меню, но в этом случае все поля заполнены данными о текущем элементе. В поле Длина введите 10, переместите курсор на поле Текущий и введите 190 / 3.

 Сохраните сделанные изменения. Нажмите клавишу F4. Для выхода без сохранения изменений используйте клавишу Esc.

Если Вы действительно изменили порядок элемента Инвент. номер, то не забудьте изменить и порядок элементов Полочный индекс и Авторский знак, в поле Текущий у них должно стоять 190 / 4 и 190 / 5 соответственно.

4. Возврат в Основное меню системы. Нажмите клавищу Esc.

#### Дополнительные возможности

Пользователь может также удалить элемент ввода данных. Для этого надо переместить курсор на элемент, подлежащий удалению, и нажать клавишу F8. Система запросит подтверждение на удаление элемента. При выборе Да элемент будет удален, при выборе Нет не будет сделано каких-либо изменений.

- NB! Удаленный таким образом элемент невозможно будет восстановить.
- NB! Но можно сделать и так, что элемент не будет удален, но ни в одном библиографическом уровне он не появится на экране при вводе или редактировании информации. Поле Уровень в меню Параметры библиографического элемента надо оставить пустым.

Можно удалить все элементы сразу. Для этого предназначены клавиши Shift+F8. Удаление будет произведено без предварительного предупреждения. Будьте внимательны. Эти клавиши удобны, если Вы создаете каталог, который не имеет никакого отношения ни к одному из существующих. Например, можно вести каталог партнеров, если у Вас в библиотеке есть Международный книгообмен, и т.д.

Можно установить фильтр для показа элементов только одного уровня. Нажмите клавиши Alt+F в меню Настройка.Библиографические элементы. Система попросит ввести библиографический уровень, для которого будет установлен фильтр. То есть, после введения m и нажатия клавиши Enter на экране будут только те элементы, у которых в поле Уровень есть m.

Можно распечатать элементы. Если Вы хотите вывести их сразу на печать, то нажмите клавиши Alt+P. Если принтер включен и горит лампочка on line, элементы будут распечатаны. Если установлен фильтр, то будут распечатаны элементы, удовлетворяющие этому фильтру.

Можно сохранить список элементов в текстовом файле. Для этого нажмите клавиши Alt+S. Система запросит имя файла для сохранения данных, введите любое имя и нажмите клавишу Enter. Если установлен фильтр, то будут сохранены элементы, удовлетворяющие фильтру.

Файл создается в директории запуска программы, например: C:\EC402\EC4\_USER\RDR\_WRT. Форматы вывода текста в файл и на печать совпадают. Выдается метка, ее название и длина.

Настройка файлов описания документов и файлов перекодировки

## Отключение файла описания документа

Если Вы считаете, что какой-то документ Вам не нужен, то Вы можете его отключить. После отключения файл описания документа сохраняется на диске, т.е. Вы всегда можете подключить его снова.

Для примера отключим документ Форма СП-1 (по названию) в каталоге Подписка

- Перейдите в каталог Подписка. Находясь в окне Каталог, переместите курсор на позицию Каталог. Нажмите клавишу F9. Используя клавишу стрелка-вниз, переместите курсор на элемент Подписка и нажмите клавишу Enter. Вы перейдете в каталог Подписка.
- 2. Войдите в режим Документы. Для этого нажмите клавишу F6.
- Отключите элемент Форма СП-1 (по названию). Для этого нажмите клавишу F8. Программа выдаст меню:

Отключить элемент Форма СП-1 (по назва? > Да < Нет

Выберите элемент Да и нажмите клавишу Enter для отключения файла. В противном случае файл не будет отключен, а Вы вернетесь в меню Документы.

- 4. Возврат в Основное меню системы. Нажмите клавишу Esc для выхода из меню Документы.
- NB! Для файлов перекодировки все то же самое, только надо нажимать клавиши Shift+F4 из окна Каталог Основного меню системы, а не F6.

Подключение уже существующего файла описания документа

В качестве примера мы подключим документ Форма СП-1 (по названию) в каталоге Подписка. В предыдущем примере мы отключили этот документ.

- 1. Перейдите в режим Документы в каталоге Подписка. Нажмите клавишу F6 из Основного меню системы.
- 2. Подключите документ. Для этого надо нажать клавишу F3. На экране меню:

-документы-

Файл: \EC402\EC4\_SUPP\DOC\.DOC Комментарий:

Помощь<F1> Coxp.<F4> Выход<Esc>

В поле Файл введите имя файла описания документа.

В нашем случае введите:

\EC402\EC4\_SUPP\DOC\SP01\_42.DOC

Нажмите клавишу Enter для перехода на следующее поле. В поле Комментарий напишите любой комментарий к документу. В дальнейшем в списке документов он будет появляться именно под этим именем. Например,

СП-1

Для сохранения введенных значений нажмите клавишу F4. К концу списка прибавится новый элемент СП-1.

3. Возврат в Основное меню системы. Нажмите клавишу Esc для выхода из меню Документы.

Если у Вас есть любой другой файл описания документа, то, зная его имя, Вы всегда можете его подключить.

NB! Для файлов перекодировки все то же самое, только надо нажимать клавиши Shift+F4 из окна Каталог Основного меню системы, а не F6.

Изменение параметров файла описания документа

Здесь будет рассмотрен пример изменения только внешних атрибутов документа, в частности, изменение имени файла описания документа и комментария к нему. Как внести простейшие изменения в сам файл описания документа, рассказано в главе Создание документов, а используемые там функции описаны в главе Файлы описания форматов экрана, документов, клавиш и файлы перекодировки.

Например, изменим имя документа Список литературы на имя Список литературы для каталога Книги , а имя файла описания документа L001\_42.DOC на LIST.DOC

- 1. Перейдите в режим Документы в каталоге Книги. Для этого нажмите клавищу F6 из Основного меню системы.
- Отредактируйте параметры документа. Нажмите клавишу F4. На экране меню, где в поле Файл – имя файла описания документа, а в поле Комментарий – название документа. Введите в поле Файл:

LIST.DOC

для перехода на следующее поле нажмите клавишу Enter. В поле Комментарий введите:

Список литературы для каталога Книги

Сохраните новые параметры, нажав клавишу F4.

!! Если файл LIST.DOC реально существует, то он будет подключен вместо старого, в противном случае он будет создан с нулевой длинной.

3. Возврат в Основное меню системы. Нажмите клавишу Esc.

- NB! Для файлов перекодировки все то же самое, только надо нажимать клавиши Shift+F4 из окна Каталог Основного меню системы, а не F6.
- NB! Настройка файлов "внешних" функций производится вручную. То

есть, создается новый файл без участия программы Библиотека 4.02, в файлах конфигурации прописываются пути и, возможно, он будет работать.

Настройка проверки на дублетность

С помощью файла ISBNCHEK.BLC в системе реализована проверка на дублетность в каталоге Книги. Но возможно его использование и в других каталогах, а также можно создать для каждого каталога свой файл проверки на дублетность, что более правильно, т.к. критерии проверки, как правило, для каждого каталога свои.

Файл ISBNCHEK.BLC - обычный файл, построенный по законам файла описания документов (см. главу Файлы описания форматов экрана, документов, клавиш и файлы перекодировки). Правда, в нем используются функции, которые не имеют никакого отношения к файлу описания документов, и они нигде не описаны в поставляемой документации.

Сейчас файл устроен таким образом, что проверка на дублетность возможна только в одном каталоге - Книги. Это неудобство можно устранить, но прежде, чем это сделать, надо подумать, по тому же критерию будут сравниваться записи в другом каталоге, или по другому.

Попробуем устроить проверку на дублетность в каталоге Подписка, и критерием дублетности будет совпадение индексов РП.

- NB!Для успешной проверки дублетов необходимо наличие поисковой таблицы по тому критерию, по которому мы собираемся сравнивать записи.
- 1. Выйдите из программы. Как нам кажется, удобнее всего работать в Нортоне.
- Создайте файл "внешних" функций для каталога Подписка, в комплекте поставки его нет. Для этого перейдите в директорию C:\EC402\EC4\_CNFW\BOOK и скопируйте файл KEY.CNF в директорию C:\EC402\EC4\_CNFW\PODP.
- 3. Отредактируйте файл KEY.CNF для каталога Подписка. Зайдите в файл любым текстовым редактором и исправьте строку

-19 {|| if(SetCW()==2.and.Set\_Help()=="MAIN", ExeFileBlock( "ISBNChek.blc" ), )

на, например:

}

-19 {|| if(SetCW()==2.and.Set\_Help()=="MAIN", ExeFileBlock( "PODPChek.blc" ), )
}

Сохраните сделанные изменения.

Таким образом, Вы указали программе, что для каталога Подписка правила проверки на дублетность находятся в файле PODPCHEK.BLC. Теперь его надо реально создать.

- 4. Создайте файл PODPCHEK.BLC, для этого перейдите в директорию C:\EC402\EC4\_SUPP\KEY, скопируйте файл ISBNCHEK.BLC в ту же самую директорию под именем PODPCHEK.BLC.
- Внесите изменения в файл PODPCHEk.BLC. Зайдите в файл любым текстовым редактором и измените метку 020а (метка ISBN) на метку 032а (метка индекса PП), а также название поисковой таблицы ISB на

IRP. Все эти изменения надо произвести в следующих строках (номер строки примерно - 34): {|| cStr := cMGetFSF("020a"),; aDubl:= FindDubl( "ISB", cStr, MEMO->CN ),; if(len(aDubl)>0,r:=chr(13)+chr(10)+padl(MEMO->CN,7)+" ",r:=""),; s := "", aeval( aDubl, SetVar(127) ),; if( !empty(s),fwrite( setvar(125), r+s ), ) } Вносить изменения надо только в первых двух строках, у Вас получится: { || cStr := cMGetFSF("032a"),; aDubl:= FindDubl( "IRP", cStr, MEMO->CN ),; if(len(aDubl)>0,r:=chr(13)+chr(10)+padl(MEMO->CN,7)+" ",r:=""),; s := "", aeval( aDubl, SetVar(127) ),; if( !empty(s),fwrite( setvar(125), r+s ), ) } Сохраните сделанные изменения. Проверка на дублетность работает по клавишам Shif+F10 из окна Каталог Основного меню каталога Подписка. Глава 4: Файлы описания форматов экрана, документов, клавиш и файлы перекодировки Основные понятия Общие представления Формат файлов Принятые соглашения Более подробно о файлах перекодировки

```
Функции получения информации из записей в формате MARC
   Несколько слов о формате MARC
   aMGetSF()
  FMALLP()
   cMGetFSF()
   FMP()
   aMGetPS()
  aMGetCard()
Функции обработки символьной информации
   Общие замечания о символьной информации
   cCutStr()
   cCutAll()
   cMakeCard()
   cColMake()
Функции для работы с массивами
  aA2to1()
   cAtoR()
   aRtoA()
   Краткое описание функций Clipper для работы с массивами
Функции общего назначения
  aeval()
  bSetBlock()
   uOverPut()
   G()
```

```
nSetBSkip()
   lEndExe()
Функции поддержки интерфейса
   uShowDMenu()
   aGetData()
  uSetDMenu()
  uSetDefault()
  uSetDSort()
  uBeginInd()
   uShowInd()
Функции, используемые только в файлах перекодировки
   Общие замечания
    SetcRcrDelim()
   SetcTagDelim()
    SetnTagSize()
    SetlKey()
    SetaKey()
    SetLVL()
   GetRecord()
   InionDeshifr()
   Replace()
   ReplTagf()
   FMP()
   FMAllP()
   DelData()
   ReplTagData()
    Split()
   ReplaArray()
   ReplData()
    InsTagfDataFt()
   DelTagfData()
   AnyShifr()
Функции, используемые в файлах "внешних" функций
    SaveREC()
   RestREC()
   Menu1()
   MakeSetPar()
   UseSet()
    Send()
Список функций программы Библиотека 4.02
   Функции получения информации из записей в формате MARC
   Функции для работы с символьной информацией
   Функции для работы с массивами
   Функции общего назначения
   Функции поддержки интерфейса
   Функции, используемые только в файлах перекодировки
   Функции, используемые в файлах "внешних" функций
```

Наиболее важные функции Clipper 5.01

### Основные понятия

Общие представления

В системе Библиотека 4.02 предусмотрена возможность создания различных форматов вывода информации на экран, программирование клавиш и перекодировка информации.

Для использования всех этих возможностей Вам необходимо иметь:

общие навыки программирования; понятие о языке управления базами данных Clipper 5.XX; представления о формате USMARC.

Каждому способу вывода информации на экран в окне Область просмотра соответствует файл описания формата экрана. Эти файлы имеют расширение FMT. Например:

A01\_42.FMT B01\_42.FMT

После установки системы Библиотека 4.02 файлы описания форматов экрана размещаются в директории \EC402\EC4\_SUPP\FMT.

Каждому способу создания документа соответствует файл описания документа. Эти файлы имеют расширение DOC. Например:

L001\_42.DOC SP01\_42.DOC

После установки системы Библиотека 4.02 файлы описания документов размещаются в директории \EC402\EC4\_SUPP\DOC.

Файлы описания "внешних" функций имеют, при поставке программы, расширение BLC (у Вас есть возможность задать свое расширение и даже создать новую директорию для их хранения). Например:

ISBNCHEK.BLC WRITEBUF.BLC

После установки системы Библиотека 4.02 файлы описания "внешних" функций размещаются в директории \EC402\EC4\_SUPP\KEY.

NB! Не забывайте, что еще есть файлы, с помощью которых задаются нестандартные правила занесения в поисковые таблицы, а также можно обрабатывать данные перед проведением поиска. Первые из этих файлов имеют расширение BLC, вторые - GET, а находятся файлы после установки системы в директории \EC402\EC4\_SUPP\BLC.

Файлы перекодировки (ФП) задают алгоритм, с помощью которого можно осуществлять:

- 1. Групповые преобразования в электронном каталоге. Например, можно изменить те или иные данные во всем каталоге или в выбранном подмножестве записей.
- 2. Преобразование информации при загрузке данных.
- 3. Преобразование информации при сохранении данных.

Файлы перекодировки имеют расширение CON. Например:

LDB\_LD1.CON

# NEWLOAD.CON

После установки системы Библиотека 4.02 файлы перекодировки размещаются в директории \EC402\EC4\_SUPP\CNV.

Любой из вышеперечисленных файлов представляет собой текстовый файл, оформленный по нижеприведенным правилам. В процессе работы эти файлы читаются программой, интерпретируются, а затем выполняются.

### Формат файлов

Файлы описания форматов экрана, документов, клавиш и файлы перекодировки являются частным случаем файлов блоков кода. Файл блоков кода это – текстовый файл, в котором содержатся:

Елоки кода Стандартное понятие языка программирования Clipper 5.01 и более поздних версий. Блоки кода начинаются и завершаются фигурными скобками {} и содержат текст, который впоследствии интерпретируется и выполняется программой.

Указатели HEADER, TEXT, COMPLETION.

Комментарии Строки, начинающиеся с символа \*.

Ниже приведен фрагмент файла блоков кода:

#### HEADER

```
* Установить значения начальных параметров
{|| if( empty( G("llDef") ), uSetDefault(), ), "" }
{|| if( empty( G("llDef") ), G("llDef", "On"),;
    nSetBSkip(3) ), "" }
{|| uSetDMenu("SRT", "On"), "" }
{|| uSetDMenu("NR", "Off"), "" }
{|| uSetDMenu("CPL",70), "" }
* Показать меню
{|| uShowDMenu() }
```

Блоки кода

Каждый файл состоит из блоков кода. Каждый блок кода (в дальнейшем просто блок) полностью соответствует понятию блока кода в языке CLIPPER 5.XX. Поэтому, чтобы разбираться с созданием файлов описания, необходимо иметь хотя бы общие представления о программировании в системе CLIPPER 5.XX.

Каждый блок имеет вид:

{|[параметры]| <действие>, ..., <действие> }

Передаваемые параметры перечисляются через запятую. Например:

{|i,s| s:=s+x, i:=i+1 }

Параметры могут отсутствовать, но вертикальные черточки (||) должны присутствовать всегда.

Под действием понимается: любой оператор присваивания,

выражение или вызов функции. Например:

{|| a:=1, b:=a+3, str(b,2)+" козы" }

Здесь: a:=1 - оператор присваивания; a+3 - выражение; str(b,2) - вызов функции.

NB! Пользуйтесь операцией присваивания в линию :=.

Синтаксис выражений внутри блока должен соответствовать синтаксису CLIPPER 5.XX.

Каждый блок в результате формирует данные. Эти данные представляют собой результат последнего действия. Так, в приведенном примере блок сформирует строку: " 4 козы".

NB! Каждый блок в файле описания ФОРМАТ ЭКРАНА или ДОКУМЕНТА должен формировать (возвращать) в результате символьную строку или несколько строк (это может быть и пустая строка - "", если данные сформированы еще не до конца). Это существенное ограничение по сравнению с общими возможностями блоков.

Блок может занимать несколько строчек. В этом случае в конце незавершенной строки ставится точка с запятой (;).

Блоки запускаются на выполнение только в момент работы программы. Поэтому "вылавливание" ошибок в написании блоков - процесс трудоемкий и, на наш взгляд, творческий. В программе предусмотрен процесс отлавливания ошибок, допущенных в блоках кода. Система в этом случае выдает сообщение:

\_\_\_\_\_ Ошибка в строке: \_\_\_\_\_\_ {|| d:=aMGetPS("852",{"b","p"},{""," "}), s:=CS(d), qout(g)}

Код ошибки: 14 Сообщение : Variable does not exist

В сообщениях такого рода ничего страшного нет, просто блок, в котором имеется ошибка, не будет выполняться, а Вы, нажав любую клавишу, можете продолжить работу.

Но, к сожалению, не все ошибки могут быть выявлены. Иногда появляется красное сообщение об ошибке, выдает его уже Clipper 5.XX, и тогда, как правило, приходится перезагружать программу "Библиотека 4.02".

NB! В блоках могут использоваться вызовы практически всех функций CLIPPER 5.XX и ряда функций системы Библиотека 4.02.

# Указатели

Указатели ограничивают разделы в файлах блоков кода. В самом общем случае файл блока кода состоит из трех разделов: заголовок, текст и заключение. Типичный его вид приведен ниже:

HEADER

┌──> \* Показать меню

```
Заго- {|| uSetDefault() }

ловок ...

> {|| uBeginInd("Ждите! Идет создание документа.") }

ТЕХТ

ТЕКТ // ОСНОВНАЯ ЧАСТЬ ОПИСАНИЯ

Текст // || aCard := aMGetCard() }

...

> {|| " " }

COMPLETION

Заключение > {|| "------" }
```

#### Заголовок

Заголовок помещается между указателями HEADER и TEXT. В заголовке содержатся блоки, которые выполняются только один раз в самом начале процесса.

- NB! Для файлов описания формата (\*.FMT) все блоки выполняются только один раз и производят действие над текущей записью. В файлах описания формата экрана ни один из указателей не используется.
- NB! Указатели можно не использовать также для файлов занесения нестандартных значений в поисковые таблицы (\*.BLC) и файлов предварительной обработки данных перед поиском (\*.GET).

## Текст

Текст помещается между указателями TEXT и COMPLETION. Он содержит блоки, которые выполняются для каждой записи, попадающей в документ или подлежащей перекодировке.

## Заключение

Заключение помещается после указателя COMPLETION. Оно содержит блоки, выполняющиеся в конце и только один раз. В нашем примере это добавление черты к концу документа.

### Комментарии

Комментариями являются строки в файлах блоков кода, которые начинаются с символа \*. При выполнении комментарии игнорируются.

NB! Комментарии НЕ МОГУТ встречаться внутри блока кода.

Результаты выполнения файлов блоков кода

При выполнении файлов блоков кода, содержащих описания формата экрана или документа, система поступает следующим образом. Если текущий блок возвращает символьную строку не нулевой длины, то эта строка добавляется к образу экрана или документа. Если блок возвращает какое-либо другое значение, то возвращаемые данные игнорируются.

NB! Предыдущее замечание относится только к файлам описания форматов экрана и документов.

Получение значений тех или иных величин внутри блока происходит либо при помощи вычислений, либо путем вызова функций. Описания

основных функций приведены ниже.

Принятые соглашения

В этом разделе будет рассказано о принятых обозначениях, которые используются в названиях функций в системе "Библиотека 4.02". Функции Clipper 5.XX далее будут везде написаны прописными буквами.

Первая буква в названии функции или переменной обозначает тип возвращаемой переменной:

- а массив;
- b блок кода;
- с строка символов ("прямоугольник текста");
- d дата;
- 1 логическая переменная (.t. или .f.);
- n число;
- и значение не определено (NIL).

Вторая буква в названии функции чаще всего обозначает группу функций:

- А функции обработки массивов;
- С функции обработки символьной информации;
- М функции для работы с записями в формате MARC;

Все заглавные буквы в названии функции, начиная со второй или третьей позиции, связаны с назначением данной функции.

Пример: aMGetSF()

имеет.

а - возвращает массив;
 М - предназначена для работы с записями в формате MARC;
 GetSF - сокращение от Get SubFields (получить подполя).

- NB! При написании имен функций в блоках кода регистр букв значения не
- NB! В функциях, которые используются только в файлах перекодировки, принятые соглашения могут и не использоваться. Смотрите по ситуации.

Более подробно о файлах перекодировки

Использование ФП при перекодировке записей ЭК

В окне "Каталог" нажмите клавиши Shift+F4. На экране появится меню "ПЕРЕКОДИРОВКА":

ПЕРЕКОДИРОВКА Подкл. <F3> Файл<F6> Редакт. <F4> Отключ. <F8> Выбор<Enter> Список<Пробел> Из 2-й в 3-ю или 4-ю по базе Из 2-й в 3-ю или 4-ю загрузк Пример загрузки из LDB файло



- F3 создание или подключение существующего файла перекодировки;
- F6 редактирование содержимого файла перекодировки;
- F4 изменение описания файла перекодировки;
- F8 отключение файла перекодировки.

Для запуска процесса перекодировки выберите необходимый файл и нажмите Enter.

#### Использование ФП при переносе информации

Подключение файла перекодировки возможно при переносе информации из одного каталога в другой.

Нажмите клавиши Shift+F5. В меню "КОПИРОВАНИЕ.УСТАНОВКА ПАРАМЕТРОВ" переместитесь на строчку "Список каталогов" и нажмите клавишу "Пробел". В меню "СПИСОК КАТАЛОГОВ" выберите каталог, в который Вы хотите перенести информацию, и нажмите клавишу Enter. На экране появится меню "ПЕРЕКОДИРОВКА". Выберите необходимый файл и нажмите Enter. После этого перед переносом информации будет отрабатываться алгоритм, заданный в файле перекодировки.

### Использование ФП при сохранении информации

Перейдите в режим "Сервис" (клавиша F2 в окне "Каталог"). Выберите в меню режим сохранения. В меню "СОХРАНЕНИЕ. УСТАНОВКА ПАРАМЕТРОВ" переместитесь на строку "Файл перекодировки:" и нажмите клавишу Пробел. На экране появится меню "ПЕРЕКОДИРОВКА". Выберите необходимый файл и нажмите Enter. После этого перед выгрузкой информации в файл сохранения будет отрабатываться алгоритм, заданный в файле перекодировки.

#### Использование ФП при загрузке информации

Перейдите в режим "Сервис" (клавиша F2 в окне "Каталог"). Выберите в меню режим загрузки. В меню "ЗАГРУЗКА. УСТАНОВКА ПАРАМЕТРОВ" переместитесь на строку "Файл перекодировки:" и нажмите клавишу Пробел. На экране появится меню "ПЕРЕКОДИРОВКА". Выберите необходимый файл и нажмите Enter. После этого перед выгрузкой информации в файл сохранения будет отрабатываться алгоритм, заданный в файле перекодировки.

NB! При загрузке информации можно задать такой алгоритм перекодировки, который позволяет загружать данные, не соответствующие требованиям формата MARC. Можно, например, загрузить информацию из обычного текстового файла. Конечно, в этом случае информация должна быть хоть как-то структурирована. Об этом смотрите раздел Функции, используемые только в файлах перекодировки.

Функции получения информации из записей в формате MARC

Несколько слов о формате MARC

Что такое запись? Записью в программе считается единица информации, зашифрованная в соответствии с форматом USMARC. С точки зрения программирования - это строка символов. Но со строкой символов неудобно работать из блоков кода. Поэтому эта информация расшифровывается, и пользователь имеет доступ к уже расшифрованной записи.

Логически расшифрованную запись можно представить так:

Метка	Данные	Признак начала поля
1008	1	.t.
100a	Чуковский, Корней Иванович	.f.
2458	0	.t.
2459	0	.f.
245a	Дневник, 1901 - 1929	.f.
260a	М.	.t.
260b	Сов.писатель	.f.
260c	1991	.f.
300a	541с.,16 л ил	.t.
300b	портр	.f.
300c	20см	.f.
500a	Указ.имен:с.506-541	.t.
653a	Чуковский,Корней Иванович	.t.
650a	СОВЕТСКАЯ ЛИТЕРАТУРА	.t.
650a	ЧУКОВСКИЙ,К.И.	.f.
650a	ДНЕВНИКИ	.f.
650a	РУССКАЯ ЛИТЕРАТУРА	.f.
020a	526501523X	.t.
020c	3.90	.f.
852b	ЧЗ	.t.
852t	3	.f.
852p	12345	.f.
852p	12346	.f.
852p	12509	.f.
852b	ΦБ	.t.
852t	1	.f.
852p	12347	.f.

В первой колонке приведен список меток. Метка состоит из кода поля (трехсимвольный код, например, 852) и кода подполя (один символ из набора: 6,8,9,a,b,c,...,x,y,z). Список всех меток для каждого каталога дан в Приложении. В дальнейшем элемент из этой колонки мы будем обозначать как переменную - cTagSF, а всю колонку - как массив - aTagSF.

Во второй колонке приведены соответствующие меткам данные. Массив данных из второй колонки будем обозначать aSASF, а переменную

из этой колонки - cSF.

Например, все инвентарные номера (подполе 852p), полученные с помощью функции aMGetSF(), соответствуют двумерному массиву:

aSASF[1][1] = 12345, aSASF[1][2] = 12346, aSASF[1][3] = 12509 aSASF[2][1] = 12347

В третьей колонке приведен признак начала поля. Значение .t. соответствует началу нового поля, значение .f. относится ко всем остальным подполям этого поля. Всю колонку будем обозначать как массив aFT, а определенное значение из нее - как переменную 1FT. aMGetSF()

Назначение Получить массив данных для заданной метки. Синтаксис aMGetSF( <cTagSF>, [<nDimension>] ) --> aSASF Использование cTaqSF Метка - строка из 4-х символов. Первые три соответствуют коду поля, четвертый - коду подполя. nDimension Размерность возвращаемого массива. Если параметр не задан или равен 2, то массив aSASF будет двумерный, при задании 1 массив aSASF будет одномерным. aSASF Одномерный или двумерный массив. Содержит данные, соответствующие метке cTagSF. Функция возвращает массив. Первый индекс массива соответствует первому подполю в поле. Второй индекс - остальным подполям в текущем поле. Пусть сигле ЧЗ соответствуют инвентарные номера: 12345, 12346, 12509, сигле АБ1 - 12347 (см. пример в предыдущем разделе). Тогла после вызова: { | | aSASF := aMGetSF( "852p" ) } получим: aSASF[1][1] = "12345", aSASF[1][2] = "12346", aSASF[1][3] = "12509" aSASF[2][1] = "12347" Получим значения инвентарных номеров в одномерный массив. { | | aSASF := aMGetSF( "852p", 1 ) } получим: aSASF[1] = "12345"aSASF[2] = "12346"aSASF[3] = "12509" aSASF[4] = "12347" Например, Вы хотите получить заглавие книги в переменную cTitle. { | | aSASF := aMGetSF( "245a" ),; cTitle := if( !empty( aSASF ), aSASF[1][1], "" ) } результат: cTitle = "Дневник, 1901 - 1929" NB!Функция empty() проверяет, не получился ли массив aSASF нулевой

NB!Функция empty() проверяет, не получился ли массив aSASF нулевой длины, т.е. есть ли в нем данные. Если массив пустой (данных о заглавии в записи нет), то при обращении к элементу массива aSASF[1][1] будет выдано сообщение об ошибке.

FMALLP()

Назначение Получить все элементы библиографического описания, соответствующие указанной метке.

Синтаксис

FMALLP( <cTag> ) --> aField

Использование

cTag Метка – строка из 4-х символов. Первые три соответствуют коду поля, четвертый – коду подполя.

aField Массив значений полей с меткой cTag.

Функция позволяет получить массив символьных строк, содержащих информацию из полей с меткой cTag. Применяется для получения всех данных при повторяющихся метках.

Получение ключевых слов.

{ || aF := FMALLP( "653a" ) }

в записи:

653а КОСМИЧЕСКИЕ АППАРАТЫ653а СПУТНИКИ653а СПУСКАЕМЫЕ АППАРАТЫ

применили:

{ || aF := FMALLP( "653a" ) }

массив аF содержит следующие элементы:

aF[1] = "КОСМИЧЕСКИЕ АППАРАТЫ" aF[2] "СПУТНИКИ" aF[3] "СПУСКАЕМЫЕ АППАРАТЫ"

NB! Функция используется только в файлах перекодировки.

```
cMGetFSF()
   Назначение
       Получить символьную строку с данными, соответствующими первому
       подполю указанной метки.
   Синтаксис
        cMGetFSF( <cTagSF> ) --> cSF
   Использование
       cTagSF
                Метка - строка из 4-х символов. Первые
                 три соответствуют коду поля, четвертый - коду
                 подполя.
           cSF
                    Символьная строка. Содержит значение подполя.
    Функция позволяет получить значение первого подполя в записи.
   Получение заглавия.
           { | | cSF := cMGetFSF( "245a" ) }
   результат:
           cSF = "Дневник, 1901 - 1929"
   Получение первой сиглы.
           { | | cSF := cMGetFSF( "852b" ) }
```

результат:

cSF = "43"

FMP()

```
Назначение
Получить элемент библиографического описания.
Синтаксис
```

FMP( <cTag> ) --> cField

Использование

cTag Метка – строка из 4-х символов. Первые три соответствуют коду поля, четвертый – коду подполя.

cField Символьная строка. Содержит значение подполя.

Функция позволяет получить символьную строку, содержащую информацию из поля с меткой сТад.

## Пример:

в записи:

 100a
 ВИЛЬЕГАС Б.Х.Э.
 .t.

 245a
 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И ЭКСПЕРТНЫЕ СИСТЕМЫ .t.

 245c
 ГОС. КОМ. СССР ПО НАР. ОБРАЗОВАНИЮ.
 .f.

 M550
 : АВТОРЕФ.ДИС. ...КАНД.ЭКОН.НАУК
 .t.

применили:

{ || cF1 := FMP( "100a" ), cF2 := FMP( "M550" ) }

в переменную cF1 "попало" значение ВИЛЬЕГАС Б.Х.Э.; в переменную cF2 "попало" значение : АВТОРЕФ.ДИС. ...КАНД.ЭКОН.НАУК

NB! Функция используется только в файлах перекодировки.

NB! В данном примере была использована запись не в формате USMARC. О загрузке данных не в формате MARC читайте в главе Загрузка информации в формате, отличном от USMARC

# aMGetPS()

Назначение Получить массив символьных строк. Каждая такая строка состоит из подполей с указанными между ними разделителями.

Синтаксис

aMGetPS( <cTag>, <aSF>, <aP>, [<lFirst>] ) --> aPS

Использование

сТад Метка поля. Трехсимвольная строка.

- aSF Одномерный массив кодов подполей, которые должны присутствовать в результирующей строке.
- аР Одномерный массив разделителей.
- lFirst Признак первого разделителя в строке.
- aPS Результат. Одномерный массив строк.

Функция позволяет получить массив строк, соответствующих полю с меткой cTag. В каждой строке приводятся подполя, коды которых указаны в массиве aSF. Перед началом каждого подполя ставится разделитель, указанный в массиве aP. Длина массивов aSF и aP должна совпадать. Если при отсутствии значений подполя, указанного в aSF первым, необходимо перед первым значащим подполем поставить разделитель, то надо задать lFirst = .t. Если этот разделитель надо опустить, то lFirst = .f. По умолчанию lFirst присваивается значение .t.

Для получения строки, соответствующей области издания, надо написать следующий кодовый блок:

результат:

cAP = "M.:Сов.писатель,1991"

Получение строки, соответствующей области ISBN.

```
{ | | aPS := aMGetPS( "020", {"a","c"}, {"ISBN ",":"}, .f. ) }
```

результат:

ISBN присутствует:
 aPS[1] = "ISBN 526501523X:3.90"
 ISBN не приведен:
 aPS[1] = "3.90"

Получение данных о хранении.

{|| aPS := aMGetPS( "852", {"b", "p"}, {"", " "} ) }

результат:

aPS[1] = "43 12345 12346 12509" aPS[2] = "AB1 12347" aMGetCard()

```
Назначение
   Получить массив, содержащий символьные строки с данными о
   заголовке, заглавии, области тома и источнике.
 Синтаксис
       aMGetCard( [<cMarc>] ) --> aCard[4]
 Использование
   сMarc Запись в формате USMARC.
       aCard
                Одномерный массив из четырех элементов.
Функция позволяет получить данные о каталожной карточке
согласно ГОСТ 7.1-84, сМагс - запись в формате USMARC.
Если этот параметр отсутствует, то данные берутся из
текущей записи.
         aCard[1] - содержит заголовок;
          aCard[2] - содержит заглавие;
          aCard[3] - содержит область данных о томе (для спецификаций);
          aCard[4] - содержит область данных об источнике.
Например:
          {|| aCard := aMGetCard() }
результат:
          aCard[1] = "Чуковский, Корней Иванович."
          aCard[2] = "Дневник, 1901-1929.-М.:Сов.писатель,
                  1991.-541с.,16 л ил:портр;20см"
          aCard[3] = ""
          aCard[4] = ""
```

Функции обработки символьной информации

Общие замечания о символьной информации

Символьная строка (строка символов) - любая последовательность символов. При использовании в блоках кодов такие строки заключаются в двойные или одинарные кавычки. Например:

cStr := "- Это ЖЖЖ не спроста!"

Текстовая строка - символьная строка, завершающаяся кодами перевода строки (chr(13)) и возврата каретки (chr(10)). Пример:

cStr := "- Это ЖЖЖ не спроста!"+chr(13)+chr(10)

Прямоугольник текста - несколько соединенных вместе текстовых строк. По сути дела, прямоугольник текста - это символьная строка, внутри которой есть символы перевода строки и возврата каретки. Пример:

> cStr := "Пух сказал:"+chr(13)+chr(10)+; "- Это ЖЖЖ не спроста!"+chr(13)+chr(10)

При выводе на экран или печать прямоугольник текста выглядит так:

Пух сказал: - Это ЖЖЖ не спроста!

Пустая строка - строка, в которой нет ни одного символа, или строка, в которой есть только символы перевода строки, возврата каретки и пробела. Примеры:

```
cStr := ""
cStr := chr(13)+chr(10)
cStr := space(5)
```

# cCutStr()

Назначение Разбить символьную строку на несколько более коротких и создать из них прямоугольник текста.

Синтаксис

cCutStr( <cStr>, <nSh1>, <nSh2>, <nWidth>, [<lDivide>] ) --> cRT

Применение

cStr Исходная строка.

- nSh1 Сдвиг первой строки в прямоугольнике текста относительно левой границы.
- nSh2 Сдвиг второй и последующих строк в прямоугольнике текста относительно левой границы.
- nWidth Определение длины строки в прямоугольнике текста.
- lDivide Установка способа переноса строк: .t. если требуется перенос внутри слов, .f. - если перенос делается только по пробелам и знакам пунктуации.
  - сRT Результат. Прямоугольник текста.

Функция создает из символьной строки прямоугольник текста. Одновременно происходит оформление абзацев.

> {|| cStr := "Я - часть той силы, что вечно хочет зла "+; "и вечно совершает благо...",; cRT := cCutStr( cStr, 3, 0, 28, .t. ) }

В результате выполнения этого кодового блока на экране в Окне Область просмотра или в документе появится следующий прямоугольник текста:

> Я - часть той силы, что вечно хочет зла и вечно совершает благо...

{|| cStr := "Я - часть той силы, что вечно хочет зла "+; "и вечно совершает благо..." cRT := cCutStr( cStr, 0, 0, 28, .f. ) }

Результат будет следующим:

Я – часть той силы, что вечно хочет зла и вечно совершает благо...

# cCutAll()

Назначение Разбить все символьные строки в массиве на несколько более коротких и создать из них прямоугольники текста.

Синтаксис

cCutAll( <aStr>, <nSh1>, <nSh2>, <nWidth>, [<lDivide>] ) --> cRT

Применение

- aStr Одномерный или двумерный массив, содержащий строки символов.
- nSh1 Сдвиг первой строки в прямоугольнике текста относительно левой границы.
- nSh2 Сдвиг второй и последующих строк в прямоугольнике текста относительно левой границы.
- nWidth Определение длины строки в прямоугольнике текста.
- lDivide Установка способа переноса строк: .t. если требуется перенос внутри слов, .f. - если перенос делается только по пробелам и знакам пунктуации.
- cRT Результат. Несколько прямоугольников текста, соединенных последовательно друг с другом.

Функция создает из каждого элемента массива aStr, являющегося символьной строкой, прямоугольник текста. Одновременно происходит оформление абзацев.

Результатом выполнения этого блока будет следующий прямоугольник текста:

...так кто ж ты, наконец? Я - часть той силы, что вечно хочет зла и вечно совершает благо...

При задании двумерного массива:

результат будет таким же, как при задании одномерного массива.

cMakeCard()

Назначение Получить прямоугольник текста, содержащий каталожную карточку по ГОСТ 7.1-84

Синтаксис

cMakeCard( <aCard> ) --> cCard

Применение

aCard Массив, содержащий строки каталожной карточки. Массив можно получить, используя функцию aMGetCard().

cCard Прямоугольник текста, содержащий каталожную карточку по гост 7.1-84.

Функция обрабатывает массив aCard, полученный с помощью функции aMGetCard(), и создает прямоугольник текста, содержащий информацию о каталожной карточке по ГОСТ 7.1-84.

Результат будет следующий:

Чуковский, Корней Иванович. Дневник, 1901-1929.-М.:Сов.писатель,1991.-541с.,16 л ил:портр;20см cColMake()

```
Назначение
   Создать прямоугольник текста, состоящий из колонок.
 Синтаксис
       cColMake( <aArray> ) --> cRect
  Применение
   aArray Массив, задающий описание колонок.
   cRect
            Прямоугольник текста.
Функция cColMake() создает прямоугольник текста, состоящий из
колонок. Описание колонок содержится в двумерном массиве aArray.
Каждая колонка описывается подмассивом aArray[i]. Подмассив
aArray[i] состоит из четырех элементов:
aArray[i][1] - содержит символьную строку, которая помещается
              в і колонку;
aArray[i][2] - содержит номер позиции левой границы і колонки;
aArray[i][3] - содержит номер позиции правой границы і колонки;
aArray[i][4] - содержит строку "L" или "R", обозначающую
               признак "прижимания" данных в колонке к левой
                ("L") или правой ("R") границам.
          {|| aCol:={}, aadd( aCol, {"Динамика машущего полета",;
             1, 10, "L"} ),;
             aadd( aCol, {"2300.00", 12, 22, "R"} ),;
             cColMake( aCol ) }
Результат на экране или принтере:
                        2300.00
          Динамика
         машущего
          полета
```

NB!В примере используются функции работы с массивами, они описываются в следующем разделе.

Краткое описание функций Clipper для работы с массивами

- aadd( <aTarget>, <expValue> ) --> Value Добавляет новый элемент expValue в конец массива aTarget.
- adel( <aTarget>, <nPosition> ) --> aTarget Удаляет элемент массива под номером nPosition из массива aTarget.
- ains( <aTarget>, <nPosition> ) --> aTarget Заносит элемент со значением NIL в массив aTarget в позицию nPosition.
- array( <nElements> ) --> aArray Создает не инициализированный массив заданной длины nElements. Возвращает массив заданной размерности.
- asize( <aTarget>, <nLength> ) --> aTarget Увеличивает или уменьшает длину массива aTarget. Новый размер массива задается величиной - nLenght.

NB!Для задания массива с не известной заранее длиной воспользуйтесь следующим выражением: а := {}. Для добавления элементов к массиву используйте функцию aadd().

aA2to1()

```
Назначение
Преобразовать двумерный массив в одномерный.
```

Синтаксис aA2tol( <aA2d> ) --> aAld

Применение аA2d Двумерный массив.

aAld Результат. Одномерный массив.

Например, Вам надо получить одномерный массив со всеми ключевыми словами.

{ || aKeys2d := aMGetSF("653a"),; aKeys1d := aA2to1( aKeys2d ) }

В результате Вы получите:

aKeys2d[1][1] = "Искусство" aKeys2d[1][2] = "Литература" aKeys2d[2][1] = "Культура" aKeys1d[1] = "Искусство" aKeys1d[2] = "Литература" aKeys1d[3] = "Культура"
cAtoR()

Назначение Преобразовать одномерный или двумерный массив в прямоугольник текста. Синтаксис cAtoR( <aArray>, [<cParse>] ) --> cRect Применение aArray Одномерный или двумерный массив, содержащий символьные строки. cParse Разделитель. Результатом. Строка, состоящая из соединенных cRect через разделитель cParse строк массива. Функция cAtoR() соединяет символьные строки одномерного или двумерного массива в "длинную" символьную строку. Между элементами массива в этой строке вставляется строка-разделитель cParse. Если cParse отсутствует, в качестве разделителя используются коды возврата каретки и перевода строки. В этом случае массив преобразуется в прямоугольник текста. Получение ключевых слов в виде колонки текста. {|| cRect := cAtoR( aMGetSF( "653a" ) ) }

Результатом будет следующий прямоугольник текста:

Искусство Литература Культура aRtoA()

```
Назначение
Преобразовать прямоугольник текста в одномерный массив.
Синтаксис
aRtoA( <cRect> ) --> aArray
```

Применение cRect - Прямоугольник текста.

aArray - Одномерный массив.

Функция aRtoA() преобразует прямоугольник текста в одномерный массив aArray, содержащий символьные строки. Каждая строка прямоугольника текста переходит в элемент массива.

Преобразование прямоугольника текста ключевых слов в массив.

```
{ | | cRect := "Искусство"+chr(13)+chr(10)+;
"Литература"+chr(13)+chr(10)+;
"Культура",;
aKeysld := aRtoA( cRect ) }
```

Результат:

aKeysld[1] = "Искусство" aKeysld[2] = "Литература" aKeysld[3] = "Культура"

## Функции общего назначения

## aeval()

Назначение Выполнение блока кода над каждым элементом массива. Синтаксис aeval( <aArr>, <bBlock>, [<nBeg>], [<nLen>] ) --> aArr Применение аArr Массив, над элементами которого выполняется bBlock. bBlock Блок кода. Выполняется над каждым элементом массива. nBeg Номер элемента массива, с которого начинается выполнение блока кода. nLen Количество элементов массива, над которыми выполняется блок кода.

В блоках кода нельзя использовать стандартные возможности Clipper для организации циклов. Но существует ряд функций, используя которые можно добиться выполнения циклов. Функция aeval() - одна из них. Функция aeval() позволяет над указанными элементами массива aArr выполнить блок кода bBlock.

Например, Вам надо суммировать элементы одномерного массива aPrice.

{|| aPrice := aMGetSF( "020a" ),; nSum = 0,; bSum = bSetBlock( "{|x| nSum := nSum + val(x) }" ),; aeval( aPrice, bSum ) }

В результате nSum содержит сумму элементов массива.

NB! Функция bSetBlock описывается в следующем разделе.

В Clipper 5.XX также есть функция eval(), она предназначена для выполнения блока кода, который задается с помощью функции bSetBlock().

eval( <bBlock>, [<BlockArg list>} ) --> LastBlockValue

bBlock Блок кода.

. . .

BlockArg list Параметры, передаваемые в блок, перечисляются через запятую.

Блок, который приведен ниже взят из файла описания документа SP01\_42.DOC (СП-1 (по названию)). Здесь показано использование всех самых сложных функций: aeval(), eval(), bSetBlock().

- \* Блок определения к-ва экземпляров. Блок определяется один
- \* раз в разделе Заголовок, а будет выполняеться для каждой записи.
- \* В переменной bl будет храниться блок кода, который определяет
- \* количество экземпляров издания для заданной сиглы. Сигла была

- \* определена выше в файле и храниться в глобальной переменной \* cSGL.
- {|| b1:=bSetBlock(;
- \* В переменной А будет храниться одномерный массив сигл.
- "{|| a:=aMGetSF('852b',1),;
- \* В переменную S получаем, определенное ранее, значение текущей \* сиглы.
- s:=G('cSGL'),;
- \* N номер текущей сиглы в массиве сигл. n:=Ascan(a,s),;
- \* В переменную D получаем массив количества экземпляров,
- \* предполагая, что каждой сигле задано количество экземпляров. d:=aMGetSF('852t',1),;
- \* В переменной SS количество экземпляров для данной сиглы. ss:=if(n>0 .and. n<=len(d),d[n],''),;
- \* Запоминаем количество экземляров в глобальную переменную cNum. G('cNum',ss) }"),;
- \* Запоминаем блок B1 в глобальную переменную bSPNum. G("bSPNum",b1) }

TEXT

. . .

```
. . .
 * Вычисляем количество экземпляров для текущей сиглы, текущей
 * записи.
 * В переменной В получаем непосредственно блок кода, который
 * должен быть выполнен, результатом его выполнения должно
 * быть определение глобальной переменной cNum, где будет
 * храниться количество экземпляров издания для сиглы, которая
 * определена в глобальной переменной cSGL.
 { | | b:=G("bSPNum"),;
 * Присваиваем глобальной переменной cNum пустое значени.
   G("cNum", "" ),;
 * Выполняем блок кода В.
   eval(b),;
 * Если количество экземляров не задано, то следующие 28 блоков
 * кода будут пропущены и выполняться не будут.
    if(Empty(G("cNum")), nSetBSkip(28), ) ,""}
    . . .
Рассмотрим использование функции aeval().
Например, если бы мы хотели подсчитать общее количество
экземляров данного издания, то в файле описания документа это
выглядело бы примерно так:
TEXT
 * В переменной А будет храниться одномерный массив количества
 * экземпляров издания. Определяем начальное значение количества
 * экземпляров в переменной N.
   {|| a:=aMGetSF('852t',1), n:=0;
 * В переменной В1 храним блок кода для подсчета общего
 количества экземпляров текущего издания.
    b1:=bSetBlock(;
 * сиглы.
```

"{|x| n := n + val(x) }"),;

- \* Посчитаем количество экземпляров.
- Aeval( a, b1),;
- \* Выдадим количество экземпляров на экран и подождем нажатия
- \* любой клавиши, для продолжения работы.
- \* qout(n), inkey(0) }

bSetBlock()

Назначение Задать блок кода. Синтаксис bSetBlock( <cStrBlock> ) --> bBlock Применение cStrBlock Строка символов, содержащая блок кода. bBlock Блок кода. Функция bSetBlock() позволяет задать блок, описанный в символьной строке cStrBlock. Для чего это надо? Дело в том, что в предыдущем примере мы использовали aeval() внутри блока кода. Там же надо определять и тот блок кода, который будет выполняться над элементами массива. Но это недопустимо! Нельзя написать что-то похожее на:  $\{ | | bSum := \{ |x| nSum := nSum + x \} \}$ В этой ситуации надо использовать функцию bSetBlock(). Вышеприведенный пример можно переписать так: {|| bSum := bSetBlock( "{|x| nSum:=nSum+x }" ) } и тогда Вы получите желаемый результат. Далее приведено несколько типичных примеров использования функций aeval() и bSetBlock(). Получить прямоугольник текста, содержащий предметные рубрики. \* Предметные рубрики {|| d:=aMGetPS("650", {"a","x"}, {""," - "}, .t. ),; b:=bSetBlock("{|x| n:=n+1, d[n]:=alltrim(str(n))+'.'; +d[n]}"), n := 0, aeval( d, b ),; cCutAll( d, 12, 9, 46 ) } В результате на экране Вы получите: 1.Чуковский, Корней Иванович 2.Советская художественная литература - детская литература Получить прямоугольник текста с ключевыми словами. \* Ключевые слова {|| d:=aA2to1( aMGetSF("653a") ),;  $b:=bSetBlock("{|x| s:=s+' - '+x}"),;$ s:="", aeval( d, b ),; s:=if( !empty(s), substr(s,4), "" ),; cCutStr(s,12,9,46,.f.) } Результат:

СОВЕТСКАЯ ЛИТЕРАТУРА – ЧУКОВСКИЙ,К. И. – ДНЕВНИКИ – РУССКАЯ ЛИТЕРАТУРА

NB!Обратите особое внимание на то, что если блок заключен в двойные кавычки (""), то переменная должна быть

в одинарных кавычках ('').

NB!Если при обработке файла блока кода программа дает сообщение об ошибке: Syntax error, в первую очередь проверьте наличие ,; или ; в конце строки, если есть перенос блока на несколько строк, а также правильное использование одинарных и двойных кавычек. uOverPut()

Назначение Перезапись информации. Синтаксис uOverPut( <cStr|aStr>, <nRow>, <nCol>, <nWidth>) --> NIL Применение cStr|aStr Символьная строка, или прямоугольник текста, или массив символьных строк. nRow Номер строки, с которой начинается перезапись информации. nCol Номер колонки, с которой начинается перезапись информации. nWidth Длина усечения данных.

Функция uOverPut() перезаписывает информацию. Перезапись начинается со строки nRow и колонки nCol. Данные для перезаписи усекаются в соответствии с nWidth.

При создании экрана или выходного документа система автоматически ведет нумерацию строк и колонок. Например:

			Кој	понк	И		
		1	2		3	4	5
	1234	567890123450	57890123	3456789	012345	5789012345	67890
С	1	Чуковся	кий <b>,</b> Корг	ней Ива	нович.		
т	2	Днен	зник,	1901-1	929M	.:Сов.писа	L —
р	3	тель,19	991542	lc.,16	л ил:по	ортр <b>;20</b> см	
0	4						
к	5						
И	б						

Номера строки и колонки можно использовать для перезаписи информации с помощью функции uOverPut().

Например, Вы хотите получить основную часть описания и перезаписать данные о полочном индексе и авторском знаке.

```
{|| cMakeCard( aMGetCard() ),;
    cStr := "7HE35"+chr(13)+chr(10)+"4-885",;
    uOverPut( cStr, 1, 1, 8 ) }
```

Результат:

5		]	Коло	нки		
		1	2	3	4	5
	12345678	9012345678	9012345	67890123	45678901234567	890
С	17HE35	Чуковский	,Корней	Иванови	ч.	
т	24-885	Дневни	к, 19	01-1929.	-М.:Сов.писа-	
р	3	тель,1991	541c.	,16 л ил	:портр;20см	
0	4					
к	5					
И	6					

Назначение Задает глобальную переменную. Синтаксис G( <cVar>, [<Val>] ) --> Val

cVar Символьная строка с именем переменной.

Val Значение переменной.

Функция G() задает глобальную переменную, имя которой cVar. Оно заносится в таблицу имен переменных, a Val - в таблицу значений. Если Val не задано, то G() возращает ранее установленное значение, если в таблице имен переменных есть переменная cVar. В случае отсутствия cVar в таблице имен переменных - возвращается NIL.

Наличие этой функции необходимо, т.к. кодовые блоки не допускают передачи значений переменных между ними. Например, при выполнении:

 $\left\{ \begin{array}{ccc} | & | & a & := & 1 , & " & " \\ | & | & a & := & a+1 , & " & " \end{array} \right\}$ 

Применение

во втором блоке возникает ошибка, т.к. значение переменной <a> в правой части выражения не определено.

Для передачи значений переменных между блоками используется функция G(). Вышеприведенный пример следует переписать так:

{ || a := 1, G("nGA",a), "" } { || a := G("nGA")+1,"" }

Результат: а=2.

NB!Имена переменных перед занесением в таблицу перетранслируются в верхний регистр.

Например,

Результат:

c = 356

G()

uSetBSkip()

```
Назначение
Пропуск выполнения нескольких блоков кода.
Синтаксис
uSetBSkip( <nSB> ) --> NIL
Применение
nSB Количество пропускаемых блоков.
```

Функция nSetBSkip() позволяет не выполнять несколько блоков. Количество пропускаемых блоков задается переменной nSB. Если nSB не задано, функция возвращает текущее значение.

Например,

```
{|| if( empty(cMGetFSF("100a"), uSetBSkip(2), "" ), "" }
{|| . . . }
{|| . . . }
{|| "Если нет данных в подполе 100а, выполняется;
    только этот блок" }
```

Строки комментариев не считаются. Например,

```
{|| uSetBSkip( 1 ) }
*{|| . . . }
*{|| . . . }
{|| "Блок будет пропущен" }
{|| "Этот блок выполнится" }
```

lEndExe()

Назначение Завершить выполнение файла блоков кода. Синтаксис lEndExe( <lEnd> ) --> lNew Применение lEnd Признак завершения выполнения: .f. - не завершать, .t. - завершить выполнение файла. lNew Новое значение признака завершения.

Функция lEndExe() предназначена для прерывания процесса создания документа. После вызова lEndExe( .t. ) блок, в котором встретилась функция, выполняется до конца, далее программа переходит к первому блоку COMPLETION. Если первым блоком поставить {|| lEndExe( .t. ) }, то остальные блоки COMPLETION выполняться не будут, в противном случае они выполняются до конца.

```
Функции поддержки интерфейса
uShowDMenu()
      Назначение
      Вызов на экран меню Документы.Установка параметров.
      Синтаксис
            uShowDMenu() --> NIL
      Применение
      Функция uShowDMenu() вызывает на экран меню Документы.Установка
      параметров для управления процессом создания документа.
      Например,
                 { | | uShowDMenu() }
        Результат - на экране появится следующее меню:
                         -документы.установка параметров-
          Имя: \EC4_PRG\WORK\EC1\LISTS.TXT
                                (□) Bce
          Источник:
                               Записи:
                                                    Номера:
             (*) Каталог
                                                     Первая
                                                                  1
              () Запрос
                                   ( ) Интервал
                                                       Последняя О
              ( ) Буфер
                                                       (0 – до конца)
              () Изменения
          Тип файла: Нумерация:
(*) Текст [X] Запис
                                            Количество:

        [X] Записей
        Символов в строке[66] 66

        [] Страниц
        Строк на странице[60] 60

              ( ) Ventura
                             [ ] Конец стр.
                                    [] Сортировка
                                   Coxp.<F4> Выход<Esc>
                      Помощь<F1>
```

Используя клавиши управления курсором и клавиши Tab, Shift+Tab, переместитесь к необходимому элементу и введите значения для текстовых и цифровых полей. Для полей типа (П) или [X] используйте Пробел для изменения значений.

Процесс создания документа начинается по клавише F4 (Сохранить установки и начать работу).

aGetData()

Назначение Ввод данных с экрана. Синтаксис aGetData( <aName>, <aValue> ) --> aNewValue Применение aName Массив символьных строк с именами величин, которые нужно ввести. aValue Массив начальных значений величин.

aNewValue Массив с новыми значениями величин.

Функция aGetData() позволяет получить путем ввода с клавиатуры данные, необходимые для внесения в документ. Все данные должны быть символьными, т.е. если Вы хотите задать число, то оно должно быть заключено в кавычки.

Например, Вы хотите получить дополнительные данные для создания инвентарной книги:

```
{ || aD := {dtoc(date()),space(5),space(4),space(20) },;
aD := aGetData( {"Дата: ","N в кн.сум.уч.: ",;
"N страницы: ","Инв.номер: "}, aD ) }
```

NB!C помощью функции date() Вы получаете дату, установленную на компьютере, а используя dtoc(), преобразуете дату в символьную переменную.

На экране следующее меню:

Заполнение элементов данных происходит по клавише Enter. Если курсор находится на последнем поле (Инв.номер), то при нажатии клавиши Enter будет завершен ввод данных и начнется создание документа. Для завершения ввода данных независимо от месторасположения курсора воспользуйтесь клавишами Ctrl+W.

В результате в массив aD заносятся данные, введенные с клавиатуры.

uSetDMenu()

Назначение Установка параметров для создания документа. Синтаксис uSetDMenu( <cPar>, [<cVal>|<nVal>]) --> <cSet>|<nSet> Применение сРаг Идентификатор, указывает один из элементов в меню Документы.Установка параметров.

cVal nVal Значение элемента.

Функция uSetDMenu() позволяет установить и/или получить значение элемента меню. Параметр cPar указывает, какой элемент устанавливается; cVal|nVal задает значение этого элемента. Если cVal|nVal отсутствует, то функция возвращает текущую установку.

Ниже приведены идентификаторы элементов и их возможные значения. В первой колонке даются названия элементов, которые обозначены в меню Документы.Установка параметров. Во второй колонкеидентификатор (cPar), в третьей указаны значения (cVal|nVal), которые может принимать идентификатор . В последней колонке даны значения, которые устанавливаются по умолчанию. Для установки значений по умолчанию служит функция uSetDefault().

Название	Идентифи-	Возможные значения	Значения по умолчанию
элемента	катор		
Имя файла	F	символьная строка	см. сноску 1
Источник	S	EC   QU   BU   CH	EC
Записи	I	All   Interval	All
Первая (запи	1СЬ) FR	ЧИСЛО	1
Последняя -'	'- LR	число	0 (до конца)
Тип файла	Т	Text   Ventura	Text
Записей	NR	On   Off	On
Страниц	NP	On   Off	Off
Конец страни	ицы ЕР	On   Off	Off
Симв.в строн	ke CPL	число	66
Строк на стр	D. LPP	число	60
Сортировка	SRT	On   Off	On

- EC Устанавливает режим создания документа с выбором записей из каталога .
- QU Устанавливает режим создания документа с выбором записей из запроса.
- BU Устанавливает режим создания документа с выбором записей из буфера.
- СН Устанавливает режим создания документа с выбором записей из списка измененных записей.
- All Задает режим, при котором из источника берутся все записи.
- Interval Задает режим, при котором из источника выбираются только записи, попавшие в интервал, указанный в пункте номера.

Text Задает тип выходного файла с переносами слов.

Ventura Задает тип выходного файла без переноса слов (в виде, удобном для дальнейшего макетирования).

On Устанавливает режим.

Off Отменяет режим.

1 Значение имени файла по умолчанию строится следующим образом: предполагается, что файл будет создан в рабочей директории текущего каталога (например, \EC402\EC4\_USER\RDR\_WRT\WORK\BOOK\); если имя файла описания документа L001\_42.DOC, то имя файла, где будет сохранен текст документа, L001\_42.TXT. То есть берется имя файла описания документа и добавляется расширение файла TXT.

Установим, например, режим создания документа из запроса со 2-й по 4-ю записи с сортировкой записей и выдадим меню на экран.

```
{|| uSetDMenu("S","QU") }
{|| uSetDMenu("I","Interval") }
{|| uSetDMenu("FR",2) }
{|| uSetDMenu("LR",4) }
{|| uSetDMenu("SRT","On") }
{|| uShowDMenu() }
```

На экран будет выдано следующее меню:

```
- ЛОКУМЕНТЫ. УСТАНОВКА ПАРАМЕТРОВ---
Имя: \EC4 PRG\WORK\EC1\LISTS.TXT
                     Записи:
Источник:
                                           Номера:
                      ( ) Bce
                                            Первая
   ( ) Каталог
                                                           2
                                             Последняя 4
                         (*) Интервал
   (*) Запрос
   () Буфер
                                               (0 – до конца)
   () Изменения
               Нумерация:
Тип файла:
                                  Количество:
   (*) Текст

        [X] Записей
        Символов в строке[66] 66

        [] Страниц
        Строк на странице[60] 60

   (*) Текса
() Ventura [] Страниц
[] Конец стр.
                                      Строк на странице[60] 60
                          [Х] Сортировка
            Помощь<F1> Coxp.<F4> Выход<Esc>
```

Получим установленные в предыдущем примере значения интервала записей.

uSetDefault()

```
Назначение
Установить параметры для создания документа по умолчанию.
Синтаксис
      uSetDefault() --> NIL
Применение
Функция uSetDefault() устанавливает значения по умолчанию. Они
приведены в описании функции uSetDMenu().
          {|| uSetDefault() }
В результате на экране появится меню Документы.Установка параметров.
Значения переменных будут установлены в соответствии с принятым
умолчанием.
            -----ДОКУМЕНТЫ.УСТАНОВКА ПАРАМЕТРОВ-----
Имя: \EC4\WORK\EC1\LISTS.TXT
Источник:
                    Записи:
                                        Номера:
   (*) Каталог
                      (*) Bce
                                          Первая
                                                     1
   () Запрос
                       ( ) Интервал
                                          Последняя О
   () Буфер
                                           (0 – до конца)
   () Изменения
Тип файла:
              Нумерация:
                                Количество:
   (*) Текст
   (*) Текст
() Ventura [] Страниц
[] Конец стр.
                [X] Записей Символов в строке[66] 66
[ ] Страниц Строк на странице[60] 60
                                  Строк на странице[60] 60
                        [Х] Сортировка
                        Coxp.<F4>
           Помощь<F1>
                                     Выход<Esc>
```

uSetDSort()

Назначение Установить параметры сортировки документа.

Синтаксис

uSetDSort( <cSet>, <cBSet> ) --> NIL

Применение cSet Идентификатор параметра сортировки.

cBSet Значение параметра.

При значении cSet, равном SRT, EXI, EXE, EXS, EXH, в качестве cBSet необходимо задавать символьные строки, содержащие описания кодовых блоков.

При значении cSet, равном OFH, задается имя файла с заголовками разделов документа.

При значении cSet, равном HDR, EMP, необходимо задавать установки "On" или "Off".

Ниже приведены описания установок.

- SRT Задает блок кода, который должен возвращать массив символьных строк - элементов сортировки; они используются при сортировке документа.
- EXI Задает блок кода, который выполняется над каждым элементом сортировки перед занесением его в таблицу сортировки.
- EXE Задает блок кода, который выполняется над каждной записью при создании документа; в качестве параметра в блок передается элемент сортировки.
- EXS Задает блок кода, который выполняется над элементом сортировки перед поиском заголовка в таблице заголовков.
- ЕХН Задает блок кода, который выполняется перед занесением заголовка в документ; в качестве параметров в блок передаются: элемент сортировки и заголовок.
- OFH Задает символьную строку, содержащую имя файла заголовков разделов (структура файла соответствует структуре внешнего словаря пользователя).
- HDR "On" или "Off"; при "On" в отсортированный документ включаются заголовки подразделов, при "Off" не включаются.
- EMP "On" или "Off"; при "On" в отсортированный документ включаются записи, в которых нет элементов сортировки, при "Off" - не включаются.

Для того, чтобы документ был отсортирован по авторам, в файл описания документа надо вставить следующие строки:

{|| uSetDSort( "SRT", "{|| aA2to1( uMGetSF('100a') ) }" ) }
{|| uSetDSort( "SRT", "{|| aA2to1( uMGetSF('700a') ) }" ) }

Для сортировки по первым трем символам авторов:

```
{|| uSetDSort( "SRT", "{|| aA2to1( uMGetSF('100a') ) }" )
{|| uSetDSort( "SRT", "{|| aA2to1( uMGetSF('700a') ) }" )
{|| uSetDSort( "EXI", "{|x| substr(x,1,3) }" ) }
Открыть файл заголовков и установить режим внесения заголовков
в документ. Заголовки разделов формируются по первому уровню
рубрик ГАСНТИ.
* Сортировка по индексам ГАСНТИ
{|| uSetDSort( "SRT", "{|| aA2to1( uMGetSF('092a') ) }" ) }
* Открытие файла заголовков
{|| uSetDSort( "OFH", "D_092.DBF" ) }
* Включение заголовка подразделов в документ
{|| uSetDSort( "HDR", "On" ) }
* Занесение в документ записей в которых нет элемента сортировки
* Они буду выходить под заголовком Неизвестный заголовок
{|| uSetDSort("EMP", "Off" ) }
* Занесение первого уровня ГАСНТИ в файл документа
{|| uSetDSort("EXI", "{|x| if( '.' $ x, substr(x,1,at('.',x)-1), x ) }" ) }
* Поиск в файле заголовков по первым двум символам
         {|| uSetDSort( "EXS", "{|x| substr(x,1,2) }" ) }
* Формирование текста заголовка
{|| uSetDSort("EXH", "{|x,y| padr( space(8)+;
   rtrim(x)+' '+rtrim(y), SetCPL() ) }" )}
Формирование строки " - То же - " в инвентарной книги.
* Сохранение строки " - То же - " в глобальной переменной aInc
                      - То же -
{|| G("aInc",{"
                                                ",;
   space(27),replicate("-",27)}),"" }
* Создание глобальной переменной cIOld
{|| G("cIOld", 0), "" }
 Следующий блок выполняется перед тем как занести
* запись в текстовый файл, т.е. после того как Вы
* подтвердили создание текстового файла. Если значение
* переменной cIOld совпадает со значением поля FieldGet(1)
* (значение текущей записи), то в документ заносится
* значение переменной aInc, а cIOld присваивается текущее значение.
{|| uSetDSort("EXE", "{|x| uOverPut( padl(x,9), 1, 14, 9),;
   if( G('cIOld')==fieldget(1),OverPutAll(G('aInc'),1,3,1,28,27), ),;
   G('cIOld',fieldget(1)) }" ), "" }
```

uBeginInd()

Назначение Показать начальное состояние меню индикации.

Синтаксис

uBeginInd(<cHeader>) --> NIL

Применение

cHeader Символьная строка, содержащая заголовок меню индикации.

Функция uBeginInd() выдает на экран начальное состояние меню индикации. Заголовок этого меню - cHeader.

Начать индикацию процесса создания документа.

uBeginInd( "Ждите! Идет процесс создания документа.")

На экране появится меню индикации:

———— Ждите! Идет процесс со	оздания документа. =====				
Г Время ————————————————————————————————————	Г Записи —————	-7			
Текущее : 16:11:45	в эк : 1				
Процесса : 00:00:00	В процессе: 1				
Завершения: тест	Пропущены : О				
с 1 Текущая запись ————————————————————————————————————	по 402				
<esc></esc>					

uShowInd() Назначение Показать текущее состояние меню индикации. Синтаксис uShowInd( <cRect | aRect> ) --> NIL Применение cRect | aRect Прямоугольник текста или массив строк. Функция uShowInd() выводит в меню индикации текущие характеристики процесса: время; время, прошедшее с начала процесса; приблизительное время завершения процесса; номер текущей записи в каталоге; номер записи в процессе; количество пропущенных (необработанных) записей. В область «Текущая запись» помещается информация, заданная значением cRet | aRet. NB!Пользоваться функцией uShowInd() можно только после вызова функции uBeginInd(). Показать текущее состояние процесса создания документа. {|| aCard = aMGetCard(),; cCard = cMakeCard( aCard ),; uShowInd( cCard ) } На экране появится меню: = Ждите! Идет процесс создания документа. =



Функции, используемые только в файлах перекодировки

Общие замечания

Рассмотрим функции на примере загрузки информации в формате, отличном от USMARC.

Сам процесс перекодировки (конвертации) данных, отличных по формату от USMARC, мало чем отличается от перекодировки данных, записанных согласно USMARC. Основная проблема – привести данные к виду, описанному в главе "Файлы описания форматов экрана и документов", раздел "Функции получения информации из записей в формате MARC", параграф "Несколько слов о формате MARC". Ниже приводится пример входного "неMARCовского" файла.

Входной файл - текстовый файл, содержащий информацию о записях в формате, отличном от USMARC. Например:

<ISN 66447 <M100 13407644 <M110 10628 <M150 045 <M500 331400 <M490 20 <M220 0809 <M310 GAVAGHAN K. <M450 TO MARKET TO MARKET... <M280 BANKING WORLD. - L., 1990. - VOL. 8, N 3. -<M250 P. 16-20. <M200 ИЗУЧЕНИЕ РЫНКА И МАРКЕТИНГОВАЯ ПОЛИТИКА БАНКОВ ВЕЛИКОБРИТАНИИ. <M210 A0691; B1410; C0671251755 <M180 БАНКИ; ИЗУЧЕНИЕ РЫНКА; МАРКЕТИНГ; ВЕЛИКОБРИТАНИЯ <M580 C</pre>

<ISN 67235 <M100 15927643 <M110 11450 <M150 045 <M500 331400 <M490 20 <M220 0810 <M310 SETH A. <M450 VALUE CREATION IN ACQUISTIONS: A REEXAMINATION OF PERFORMANCE ISSUES <M280 STRATEGIC MANAGEMENT J. - CHICHESTER ETC., 1990. - VOL.</pre> 11, N 2. -<м200 СОЗДАНИЕ НОВОЙ РЫНОЧНОЙ СТОИМОСТИ В РЕЗУЛЬТАТЕ ПОГЛОЩЕНИЙ И СЛИЯНИЙ. АНАЛИЗ С УЧЕТОМ ЭКОНОМИЧЕСКИХ ХАРАКТЕРИСТИК ФИРМ-УЧАСТНИЦ КОНГЛОМЕРАТОВ: ОБЪЕМ ПРОИЗВОДСТВА, ТЕХНОЛОГИЯ, ФИНАНСОВЫЕ РЕСУРСЫ, ТРУДОВЫЕ РЕСУРСЫ, МАРКЕТИНГ И т.д. <M210 A068117; A06572726 <м180 ПОГЛОЩЕНИЯ И СЛИЯНИЯ; КОМПАНИИ; КАПИТАЛИСТИЧЕСКОЕ ПРЕДПРИЯТИЕ; ТЕХНИЧЕСКИЙ УРОВЕНЬ ПРОИЗВОДСТВА; ОБЪЕМ ПРОИЗВОДСТВА; ФИНАНСОВЫЕ РЕСУРСЫ; МАРКЕТИНГ; КАДРОВАЯ ПОЛИТИКА <M580 M

Запись – логическая единица информации во входном файле. По сути дела, входной файл состоит из записей. В вышеприведенном примере входного файла содержится две записи. Первая из них:

<ISN 66447 <M100 13407644 <M110 10628 <M150 045 <M500 331400 <M490 20 <M220 0809 <M310 GAVAGHAN K. <M450 TO MARKET TO MARKET... <M280 BANKING WORLD. - L., 1990. - VOL. 8, N 3. -<M250 P. 16-20. <M200 ИЗУЧЕНИЕ РЫНКА И МАРКЕТИНГОВАЯ ПОЛИТИКА БАНКОВ ВЕЛИКОБРИТАНИИ. <M210 A0691; B1410; C0671251755 <M180 БАНКИ; ИЗУЧЕНИЕ РЫНКА; МАРКЕТИНГ; ВЕЛИКОБРИТАНИЯ <M580 C</pre>

Поле - часть записи. Соответствует понятию области библиографического описания. Например:

<M280 BANKING WORLD. - L., 1990. - VOL. 8, N 3. -

Метка поля - строка символов в начале поля (или в справочнике), определяющая, какая информация содержится в поле. Например:

<M280

Подполе - часть поля. Соответствует понятию библиографического элемента. Например:

, 1990.

Разделитель - последовательность символов, отделяющих одну запись от другой, или одно поле от другого, или одно подполе от другого подполя. Например, запятая в предыдущем примере.

Ниже разбирается пример, в котором данные приводятся к стандартному виду, а затем преобразуются в запись в формате MARC.

Приведем типичный пример файла перекодировки:

```
HEADER
            {|| SetcRcrDelim( chr(13)+chr(10)+chr(13)+chr(10) ) }
Заго-
             {|| SetcTagDelim( "<" ) }</pre>
ловок
             { | SetnTagSize(5) }
            TEXT
             { || SetLVL( "m" ) }
               cStr := GetRecord(),;
                  cStr := StrTran( cStr, chr( 13 ) + chr( 10 ), " " ),;
                  cStr := StrTran( cStr, chr( 13 ), "" ),;
Текст
                  cStr := StrTran( cStr, chr( 10 ), "" ),;
                  InionDeShifr( cStr ) }
             { || if( IsTagf( "M280" ), SetLVL( "b" ), ) }
{ || DelData( { "ISN", "M110" } ) }
{ || Replace( "M310", "100a" ) }
            {
          -> COMPLETION
Заклю-
             { || qout("Все Ok. Для продолжения нажмите любую клавишу"),;
чение
                  inkey(0) }
```

Заголовок помещается между указателями HEADER и TEXT. Он содержит кодовые блоки, определяющие: разделители записей и полей; коды первых символов подполей; длину меток, разделителей, индикаторов.

Текст помещается между указателями TEXT и COMPLETION. Он содержит кодовые блоки, которые выполняются для каждой записи из входного файла.

Заключение помещается после указателя COMPLETION. Оно содержит

кодовые блоки, выполняющиеся в конце расшифровки.

SetcRcrDelim()

Назначение Установить (получить) разделитель записи.

Синтаксис

SetcRcrDelim( [<cDelim>] ) --> cDelim

Применение

cDelim Строка символов разделителя.

Функция позволяет установить разделитель между записями во входном файле. Если параметр cDelim не задан, то возвращается его текущее значение.

Пример:

{ | | SetcRcrDelim( chr(13)+chr(10)+chr(13)+chr(10) ) }

В данном примере устанавливается разделитель, согласно которому записи разделяются пустой строкой. Комбинация символов chr(13)+chr(10) для текстовых файлов DOS означает конец строки, а ее повторение - одну пустую строку.

```
SetcTagDelim()
  Назначение
     Установить (получить) разделитель поля.
   Синтаксис
      SetcTagDelim( [<cDelim>] ) --> cDelim
   Применение
      cDelim
              Строка символов разделителя.
   Функция позволяет установить разделитель между полями. Если параметр
   cDelim не задан, то возвращается его текущее значение.
   Пример:
      {|| SetcTagDelim( "<" ) }</pre>
   При такой установке в части записи, приведенной ниже:
            . . .
      <M500 331400 <M490 20 <M220 0810
      <M310 SETH A.
            . . .
   будет выделено четыре поля:
      M500 331400
      M490 20
      M220 0810
      M310 SETH A.
```

```
SetnTagSize()
```

```
Назначение
Установить (получить) длину метки поля.
Синтаксис
{|| SetnTagSize( [<nLen>] ) } --> nLen
Применение
nLen Длина метки поля.
Функция позволяет установить длину метки поля. Если параметр
nLen не задан, то возвращается его текущее значение.
Пример:
{|| SetnTagSize( 5 ) }
```

В поле:

M310 SETH A.

пять первых символов, включая пробел, играют роль метки.

SetlKey()

Назначение Устанавливает признак записи в поисковые таблицы. Синтаксис SetlKey( [<lWriteTable>] ) --> lWriteTable Применение

lWriteTable Признак записи информации в поисковые таблицы.

Функция позволяет задать или отменить процесс разнесения поисковых элементов по поисковым таблицам. При lWriteTable = .t. одновременно с расшифровкой информации происходит обновление поисковых таблиц. При lWrite = .f. информация в поисковые таблицы не заносится. Если параметр lWrite не задан, то возвращается его текущее значение.

Пример:

```
{ || SetlKey( .t. ) }
```

Задает режим обновления поисковых таблиц.

SetaKey()

```
Назначение
Задает массив номеров обновляемых поисковых таблиц.
```

```
Синтаксис
SetaKey( [<aNumber>] ) --> aNumber
```

Применение

aNumber Массив номеров поисковых таблиц.

Функция задает номера поисковых таблиц, в которых будет производиться обновление одновременно с расшифровкой информации.

Пример:

```
{ || SetaKey( {1,2,3} ) }
```

Информация будет обновляться в 1,2 и 3 таблицах.

SetLVL()

```
Назначение
Установить (получить) библиографический уровень.
```

```
Cuhtakcuc
SetLVL( [<cLevel>] ) --> cLevel
```

Применение

cLevel Библиографический уровень записи.

Функция задает (возвращает) библиографический уровень документа. Список библиографических уровней, принятых в системе, приведен ниже:

```
а: Глава (часть,раздел,...) книги, диссертации,
b: Статья из периодического издания
c: Коллекция, библиотека, собрание, ...
d: Часть коллекции, библиотеки, собрания, ...
m: Монография ( книга, диссертация, ... )
s: Периодическое издание.
```

Пример:

```
{ || SetLVL( "m" ) }
```

Установить уровень m (монография).

GetRecord()

```
Назначение
Получить текущую запись из входного файла.
```

Синтаксис GetRecord() --> cRecord

Применение

cRecord() Текущая запись из входного файла.

Функция возвращает текущую запись во входном файле как символьную строку.

Пример:

```
{ || cRecord := GetRecord(),;
     cRecord := StrTran( cRecord, chr( 13 ) + chr( 10 ), " " ),;
```

В примере текущая запись из входного файла "попадает" в переменную cRecord. Затем в cRecord с помощью стандартной функции StrTran все коды конца строки заменяются на пробелы.

InionDeshifr()

Назначение Расшифровка записи. Синтаксис InionDeshifr( <cRecord> )

Применение cRecord Текущая запись.

Функция расшифровывает запись cRecord и помещает полученные данные в таблицу. Логически эту таблицу можно представить в следующем виде:

Метка	Данные
М310	SETH A.
М450	VALUE CREATION IN ACQUISTIONS: A REEXA
M280	STRATEGIC MANAGEMENT J CHICHESTER ETC., 1990

Ряд функций, приведенных ниже, позволяет изменять эту таблицу, задавая различные режимы перекодировки данных.

Пример:

{ || InionDeShifr( cRecord ) }

Из записи:

<ISN 66447 <M100 13407644 <M110 10628 <M150 045 <M500 331400 <M490 20 <M220 0809 <M310 GAVAGHAN K. <M450 TO MARKET TO MARKET... <M280 BANKING WORLD. - L., 1990. - VOL. 8, N 3. -<M250 P. 16-20. <M200 ИЗУЧЕНИЕ РЫНКА И МАРКЕТИНГОВАЯ ПОЛИТИКА БАНКОВ ВЕЛИКОБРИТАНИИ. <M210 A0691; B1410; C0671251755 <M180 БАНКИ; ИЗУЧЕНИЕ РЫНКА; МАРКЕТИНГ; ВЕЛИКОБРИТАНИЯ <M580 C</pre>

получим таблицу:

```
Метка Данные Признак начала поля
```

ISN 66447 .t. M100 13407644 M110 10628 .t. .t. M150 045 M500 331400 M490 20 .t. .t. .t. M220 0809 .t. M310 GAVAGHAN K. . t. . M450 TO MARKET TO MARKET... .t. M280 BANKING WORLD. - L., 1990. - VOL. 8, N 3. -.t. M250 P. 16-20. .t. M200 ИЗУЧЕНИЕ РЫНКА И МАРКЕТИНГОВАЯ ПОЛИТИКА БАНКОВ... .t. M210 A0691; B1410; C0671251755 .t. М180 БАНКИ; ИЗУЧЕНИЕ РЫНКА; МАРКЕТИНГ; ВЕЛИКОБРИТАНИЯ .t.

NB! InionDeshifr() используется для расшифровки всех файлов в формате, отличном от формата MARC. Если необходимо расшифровать данные в формате MARC следует использовать функцию MARCDeshifr(). MarcDeshifr()

Назначение Расшифровка записи в формате USMARC.

Синтаксис MarcDeshifr( <cRecord> )

Применение cRecord Текущая запись в формате USMARC.

Функция расшифровывает запись cRecord и помещает полученные данные в таблицу. Логически эту таблицу можно представить так же, как в описании предыдущей функции.

NB! Для получения текущей записи каталога используйте функцию cMGet(). Например, cRecord := cMGet()

Replace()

Назначение Заменить метку библиографического элемента.

Синтаксис Replace( cTagIn, cTagOut )

Применение

cTagIn Метка, уже существующая в таблице.

cTagOut Метка, на которую надо заменить метку cTagIn

Функция позволяет изменить значения меток в таблице. Используется для приведения расшифрованных данных к виду формата USMARC.

Пример:

Было в таблице:

M220	0809	.t.
M310	GAVAGHAN K.	.t.
M450	TO MARKET TO MARKET	.t.

применили функцию:

{ || Replace( "M310", "100a" ) }

Получилось:

M220	0809	.t.
100a	GAVAGHAN K.	.t.
M450	TO MARKET TO MARKET	.t.

ReplTagf()

Назначение Заменить метку библиографического элемента с размещением в нужном месте таблицы.

Синтаксис

ReplTagf( cTagIn, cTagOut )

Применение

cTagIn Метка, уже существующая в таблице.

cTagOut Метка, на которую надо заменить метку cTagIn.

Функция позволяет перекодировать значения меток в таблице. При этом происходит объединение подполей записи формата USMARC в поля.

Пример:

Было в таблице:

 100a
 ВИЛЬЕГАС Б.Х.Э.
 .t.

 M450
 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И ЭКСПЕРТНЫЕ СИСТЕМЫ .t.

 M550
 : АВТОРЕФ.ДИС. ...КАНД.ЭКОН.НАУК
 .t.

 M560
 ГОС. КОМ. СССР ПО НАР. ОБРАЗОВАНИЮ.
 .t.

применили:

{ || ReplTagf( "M450", "245a" ) }
{ || ReplTagf( "M560", "245c" ) }

получили:

 100a
 ВИЛЬЕГАС Б.Х.Э.
 .t.

 245a
 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И ЭКСПЕРТНЫЕ СИСТЕМЫ .t.

 245c
 ГОС. КОМ. СССР ПО НАР. ОБРАЗОВАНИЮ.
 .f.

 M550
 : АВТОРЕФ.ДИС. ...КАНД.ЭКОН.НАУК
 .t.

FMP()

```
Назначение
     Получить элемент библиографического описания.
   Синтаксис
     FMP( <cTag> ) --> cField
   Применение
                  Метка поля в таблице.
      cTag
      cField
                  Значение поля с меткой сТад.
Функция позволяет получить символьную строку, содержащую информацию
из поля с меткой cTag.
Пример:
в таблице:
   100а ВИЛЬЕГАС Б.Х.Э.
                                                         .t.
   245а ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И ЭКСПЕРТНЫЕ СИСТЕМЫ .t.
   245с ГОС. КОМ. СССР ПО НАР. ОБРАЗОВАНИЮ.
                                                         .f.
```

M550 : АВТОРЕФ.ДИС. ...КАНД.ЭКОН.НАУК .t.

применили:

{ || cF1 := FMP( "100a" ), cF2 := FMP( "M550" ) }

в переменную cF1 "попало" значение ВИЛЬЕГАС Б.Х.Э.; в переменную cF2 "попало" значение : АВТОРЕФ.ДИС. ...КАНД.ЭКОН.НАУК FMALLP()

```
Назначение
Получить все элементы библиографического описания, соответствующие
указанной метке.
```

```
Синтаксис
FMALLP( <cTag> ) --> aField
```

Применение сТад

Метка поля в таблице.

aField Массив значений полей с меткой cTag.

Функция позволяет получить массив символьных строк, содержащих информацию из полей с меткой cTag. Применяется для получения всех данных при повторяющихся метках.

Пример:

в таблице:

653а КОСМИЧЕСКИЕ АППАРАТЫ653а СПУТНИКИ653а СПУСКАЕМЫЕ АППАРАТЫ

применили:

{ || aF := FMALLP( "653a" ) }

массив aF содержит следующие элементы:

aF[1] = "КОСМИЧЕСКИЕ АППАРАТЫ" aF[2] "СПУТНИКИ" aF[3] "СПУСКАЕМЫЕ АППАРАТЫ"
DelData()

Назначение Удалить элемент библиографического описания. Синтаксис DelData( <aTagDel> ) Применение aTadDel Массив меток, которые подлежат удалению. Функция удаляет из таблицы метки и данные, соответствующие меткам в массиве aTagDel. Пример: в таблице: M500 331400 .t. M490 20 .t. M220 0809 .t. M310 GAVAGHAN K. M450 TO MARKET TO MARKET... .t.

применили:

{ || aDT := { "M490", "M220" }, DelData( aDT ) }

м500	331400	.t.
M310	GAVAGHAN K.	.t.
M450	TO MARKET TO MARKET	.t.

# ReplTagData()

Назначение Изменить библиографические данные, соответствующие указанной метке.

### Синтаксис

ReplTagData( <cTag>, <cData> )

Применение

cTag Метка поля, в котором заменяются данные.

cData Данные.

Функция позволяет поменять информацию в таблице в поле с меткой сТад.

## Пример:

#### в таблице:

М500	331400	.t.
M310	GAVAGHAN K.	.t.

## применили:

{ || ReplTagData( "M500", "000000" ) }

M500	000000	.t.
M310	GAVAGHAN K.	.t.

Split()

Назначение Разбить библиографический элемент на несколько новых библиографических элементов. Синтаксис

Split( <aDelim>, <aTag>, <aFT> )

Применение

aDelim Массив разделителей

aTag Массив меток, на которые разбиваются данные.

аFT Массив признаков поля/подполя.

Функция ищет в таблице первую метку, указанную в массиве aTag, и разбивает ее на подполя по разделителям, указанным в массиве aDelim. В массиве aFT указывается логическое значение, в соответствии с которым текущее поле является либо подполем предыдущего, либо началом нового.

Пример:

в таблице:

260а М.:Мир,1993 г. .t.

применили:

260a	Μ.		.t.
260b	Мир		.f.
260c	1993	г.	.f.

ReplaArray()

Назначение Разбить библиографический элемент на несколько элементов с такой же меткой.

Синтаксис

ReplaArray( <cTag>, <cDelim>, <lFT> )

Применение

сТад Метка поля.

cDelim Строка разделителя.

lFT Признак начала нового поля.

Функция разбивает данные в поле с меткой сТад на подполя (или поля) с той же самой меткой в соответствии с разделителем cDelim. Если lFT = .t., каждое новое значение считается новым полем. Если lFT = .f., каждое новое значение считается подполем предыдущего.

Пример:

в таблице:

653а КОСМИЧЕСКИЕ АППАРАТЫ; СПУТНИКИ; СПУСКАЕМЫЕ АППАРАТЫ .t.

применили:

{ || ReplaArray( "653a", ";", .f. ) }

653a	КОСМИЧЕСКИЕ	С АППАРАТЫ	.t.
653a	СПУТНИКИ		.f.
653a	СПУСКАЕМЫЕ	АППАРАТЫ	.f.

ReplData()

Назначение Заменить одни данные на другие. Синтаксис ReplData( <cTag>, <cDataOld>, <cDataNew> ) Применение cTag Метка поля. cDataOld Данные, которые нужно заменить. cDataNew Данные, на которые заменяется cDataOld.

Функция ищет поля, соответствующие метке cTag, проверяет все данные на равенство строке cDataOld, и если находит таковые, то заменяет на строку cDataNew.

Пример:

в таблице:

 653a
 КОСМИЧЕСКИЕ АППАРАТЫ
 .t.

 653a
 СПУТНИКИ
 .f.

 653a
 СПУСКАЕМЫЕ АППАРАТЫ
 .f.

применили:

{ || ReplData( "653a", "СПУТНИКИ", "СПУТНИК" ) }

получили:

 653a
 КОСМИЧЕСКИЕ АППАРАТЫ
 .t.

 653a
 СПУТНИК
 .f.

 653a
 СПУСКАЕМЫЕ АППАРАТЫ
 .f.

InsTagfDataFt()

Назначение Добавить новое поле/подполе вместе с данными.

Синтаксис

InsTagfDataFt( <cTag>, <cData>, <lFt> )

Применение

cTag	Метка по	ля.		
cData	Данные.			
lFt	Признак	начала	нового	поля.

Функция добавляет новое поле вместе с данными в таблицу. Если lFT := .t., то новая метка трактуется как новое поле. Если lFT :=.f., то новая метка трактуется как подполе уже существующего поля.

Пример:

в таблице:

653a	КОСМИЧЕСКИЕ	АППАРАТЫ	.t.
653a	СПУТНИКИ		.f.
653a	СПУСКАЕМЫЕ	АППАРАТЫ	.f.

применили:

{ || InsTagfDataFT( "653a", "ЗОНДЫ", .f. ) }

653a	КОСМИЧЕСКИЕ АППАРАТЫ	.t.
653a	СПУТНИК	.f.
653a	СПУСКАЕМЫЕ АППАРАТЫ	.f.
653a	ЗОНДЫ	.f.

DelTagfData() Назначение Удаляет данные из поля, если сами данные равны указанной строке. Синтаксис DelTagfData( <cTag>, <cData> ) Применение cTag Метка поля. cData Данные, которые необходимо удалить. Функция ищет поля, соответствующие метке сТад, проверяет все данные на равенство строке cData, и если находит таковые, то удаляет их. Пример: в таблице: 653а КОСМИЧЕСКИЕ АППАРАТЫ .t. 653а СПУТНИКИ .f. 653а СПУСКАЕМЫЕ АППАРАТЫ .f. применили: { || DelTagfData( "653a", "CПУТНИКИ" ) }

653a	КОСМИЧЕСКИ	Е АППАРАТЫ	.t.
653a	СПУСКАЕМЫЕ	АППАРАТЫ	.f.

AnyShifr()

Назначение Зашифровка записи в отличном от USMARC формате.

Синтаксис

AnyShifr( <cType> )

Применение

сТуре Параметр, значение которого может быть только GPNTBINUMARC.

Функция зашифровывает текущую запись в формате USMARC, если она используется без параметров; если используется параметр, то Вы получите запись в формате UNIMARC библиотеки ГПНТЕ.

Ряд перечисленных выше функций позволяет получить элементы расшифрованной записи как символьную строку, либо как массив символьных строк. Затем с ними можно провести ряд операций, используя стандартные функции Clipper 5.01. Смотрите раздел документации Наиболее важные функции Clipper 5.01. Для более подробной информации обращайтесь к документации по Clipper 5.01 и Clipper Tools II.

### Функции, используемые в файлах "внешних" функций

Прежде чем перейти к самим функциям, приведем с подробными комментариями один из таких файлов - WRITEBUF.BLC, с помощью которого в программе реализована возможность занесения в буфер не по одной записи, а сразу заданное множество записей (клавиши Alt+F5 в Основном меню системы).

Содержимое файла WRITEBUF.BLC:

```
* Запись множества в буфер
     HEADER
     * Сохранить 23 строку экрана в переменной cSCR
     { | | G( "cSCR", SaveScreen( 23, 0, 23, 79 ) ) }
     * Сохранить положение курсора в переменной cCUR
     { || G( "cCUR", SetCursor() ) }
     * Запомнить текущее положение в электронном каталоге
     \{ | | SaveREC() \}
     * Стереть 23 строку
     { | setpos(23, 0), qqout(space(80)) }
     * Вызвать меню 1 типа, получить параметры множества,
     * задать множество В1
     {|| a:=Menu1("ЗАПИСЬ В БУФЕР.УСТАНОВКА ПАРАМЕТРОВ"),;
         d:=MakeSetPar( a[1], a[2], a[3] ),;
         UseSet( "B1",d[1], d[2], d[3], d[4] ) }
     * Показать меню индикации
     {|| if( ! lEndExe(), uBeginInd("Запись в буфер."),;
        RestScreen( 23, 0, 23, 79, G("cSCR") )) }
     TEXT
     * Далее все операции выполняются над записями, заданными
     * множеством В1.
     {|| uShowInd( "Запись N: "+str(MEMO->CN) ) }
       Записать текущую запись в буфер
     {|| WriteBUFF( MEMO->CN, "SET" ) }
     COMPLETION
     * Восстановить положение курсора
     { || setcursor( G("cCUR") ) }
      Очистить множество
     { | | ClearSet() }
     * Вернуться на прежнее место в электронном каталоге
     \{ | | RestREC() \}
      Восстановить 23 строку
     { | | RestScreen( 23, 0, 23, 79, G( "cSCR" ) ) }
     NB! В файле встречаются как функции Библиотека 4.02, так
         и функции Clipper, их описание смотрите в главе Наиболее
         важные функции Clipper 5.01. Далее приводятся функции только
         системы Библиотека 4.02.
SaveREC() запоминает положение в электронном каталоге.
RestREC() восстанавливает положение в электронном каталоге.
Menul()
           показывает на экране меню вида:
             --ЗАПИСЬ В БУФЕР.УСТАНОВКА ПАРАМЕТРОВ-
   Источник:
                      Записи:
                                          Номера:
     (*) Каталог
                       (*) Bce
                                          Первая
                                                      1
      ( ) Запрос
                         ( ) Интервал
                                            Последняя О
      ( ) Буфер
                                             (0 – до конца)
```

```
() Изменения
```

Пример: a:=Menu1("ЗАПИСЬ В БУФЕР.УСТАНОВКА ПАРАМЕТРОВ") Функция возвращает массив, содержащий текущие установки. Затем этот массив может использоваться в функции MakeSetPar() для получения данных о множестве записей. MakeSetPar() преобразует данные, полученные из массива параметров меню (функция Menul() ), в параметры для задания множества записей. Пример: d:=MakeSetPar( a[1], a[2], a[3] ) где a[1], a[2], a[3] - элементы массива, полученного с помощью функции Menul(). Массив d содержит информацию о множестве записей. Его элементы необходимо использовать в функции UseSet() UseSet() устанавливает режим перемещения по множеству. UseSet( <cSet>, <cS>, <lAll>, <nF>, <nL> ) где cSet - имя множества - источник: ЕС | QU | BU | CH сS lAll - флаг; если lAll = .t., то перемещение происходит по всем записям источника, если IAll = .f., то только по интервалу, заданному nF и nL nF - первая запись в интервале nL - последняя запись в интервале Пример: UseSet("S1","EC",.f.,3,7) Задать множество S1 для перемещения по каталогу в интервале записей с 3 по 7 После использования функции UseSet() действия, указанные после разделителя ТЕХТ в файле блоков кода, будут выполняться над записями множества. Send( aArray ) Функция используется только в файлах, которые задают нестандартные поисковые таблицы, а также при предварительной обработке информации перед проведением

aArray Обязательно MACCИB значений, которые или

файлах ничего не будет.

поиска. Если этой функции нет, то работать в этих

заносятся в поисковую таблицу (каждое отдельно), или значение, по которому будет произведен поиск. В файлах с расширением GET это условно массив, т.к. там для поиска всегда передается одно значение. Если значение не определено, то напишите - Send( {""}). Список функций программы Библиотека 4.02

Функции получения информации из записей в формате MARC

- aMGetSF() Получить массив всех данных, соответствующих данной метке. В файлах перекодировки используется функция FMALLP().
- cMGetFSF() Получить символьную строку, соответствующую первым данным указанной метки. В файлах перекодировки используется функция FMP().
- aMGetPS() Получить массив символьных строк. Каждая такая строка состоит из данных, соответствующих указанным кодам полей и подполей с заданными разделителями между ними.
- aMGetCard() Получить массив, содержащий символьные строки с данными о заголовке, заглавии, области тома и источнике.

Функции обработки символьной информации

- cCutStr() Разбить символьную строку на несколько более коротких и создать из них прямоугольник текста.
- cCutAll() Разбить все символьные строки в массиве на несколько более коротких и создать из них прямоугольники текста.
- cMakeCard() Получить прямоугольник текста, содержащий каталожную карточку по ГОСТ 7.1-84.
- cColMake() Создать прямоугольник текста, состоящий из колонок.

Функции для работы с массивами

- aA2to1() Преобразование двумерного массива в одномерный.
- cAtoR() Создать из одномерного (или двумерного) массива прямоугольник текста.
- aRtoA() Преобразовать прямоугольник текста в одномерный массив.

#### Функции общего назначения

В этом разделе перечисляются функции, которые трудно отнести куда-либо еще. Это функции, позволяющие организовать выполнение циклов внутри блоков кода, перезаписать информацию, и некоторые другие.

- aeval() Выполнить блок кода над каждым элементом массива.
- bSetBlock() Задать блок кода.
- eval() Выполнить блок кода.
- uOverPut() Перезаписать информацию.

G() Задать глобальную переменную.

uSetBSkip() Пропустить выполнение нескольких блоков.

lEndExe() Завершить выполнение файла блоков кода.

Функции поддержки интерфейса

- uShowDMenu() Показать меню установки параметров для создания документа.
- aGetData() Ввод данных с экрана.
- uSetDefault() Установить параметры для создания документа по умолчанию.
- uSetDMenu() Установить параметры для создания документа.
- uSetDSort() Установить параметры, определяющие режим сортировки документа.
- uBeginInd() Показать начальное состояние меню индикации.
- uShowInd() Показать текущее состояние меню индикации.

Функции, используемые только в файлах перекодировки

- SetcRcrDelim() Установить (получить) разделитель записи.
- SetcTagDelim() Установить (получить) разделитель поля.
- SetnTagSize() Установить (получить) длину метки поля.
- SetlKey() Устанавливает признак записи в поисковые таблицы.
- SetaKey() Задает массив номеров обновляемых поисковых таблиц.
- SetLVL() Установить (получить) библиографический уровень.
- GetRecord() Получить текущую запись из входного файла.
- InionDeshifr() Расшифровка записи.
- Replace() Заменить метку библиографического элемента.
- ReplTagf() Заменить метку библиографического элемента с размещением в нужном месте таблицы.
- DelData() Удалить элемент библиографического описания.
- ReplTagData() Изменить библиографические данные, соответствующие указанной метке.
- Split() Разбить библиографический элемент на несколько новых библиографических элементов.
- ReplaArray() Разбить библиографический элемент на несколько элементов с такой же меткой.
- ReplData() Заменить одни данные на другие.
- InsTagfDataFt() Добавить новое поле вместе с данными.

DelTagfData() Удаляет данные из поля, если сами данные равны указанной строке.

AnyShifr() Зашифровка записи в отличном от USMARC формате.

Функции, используемые в файлах "внешних" функций

- SaveREC() Запоминает положение в электронном каталоге.
- RestREC() Восстанавливает положение в электронном каталоге.
- Menul() Показывает на экране меню вида "Установка параметров".
- MakeSetPar() Преобразует данные, полученные из массива параметров меню (функция Menul()), в параметры для задания множества записей.
- UseSet() Устанавливает режим перемещения по множеству.
- Send() Передача в программу данных.
- NB! Функции, про которые сказано, что они используются только в конкретных файлах, можно попробовать использовать и в других файлах. А вдруг работает?

Наиболее важные функции Clipper 5.01

При создании блоков кода для обработки символьной информации можно пользоваться стандартными функциями Clipper 5.01. Ниже приведено краткое описание наиболее важных из них. Для подробной информации смотрите описание Clipper. !Первая буква в названии переменной указывает ее тип. массив; а символьная переменная; С n число; ехр выражение; может быть числом, строкой, а иногда и массивом. alltrim( <cString> ) --> cTrimString Удаляет ведущие и замыкающие пробелы в строке символов cString. at( <cSearch>, <cTarget> ) --> nPosition Возвращает позицию вхождения подстроки cSearch в строке cTarget. chr( <nCode> ) --> cChar Преобразует код ASCII (nCode) в его символьное значение. Возвращает строку из одного символа. empty( <exp> ) --> lEmpty Определяет, не является ли выражение (массив, строка или число) пустым. Возвращает .Т. (если пустое) или .F. if( <lCondition>, <ExpTrue>, <ExpFalse> ) --> lValue Возращает данные в зависимости от результатов проверки условия. Примеры: a := 1 s := if( a == 1, "Коза", "Козел" ) в результате в в попало слово Коза a := 2 s := if( a == 1, "Коза", "Козел" ) в результате в в попало слово Козел len( <cString aTarget> ) --> nCount Возвращает длину символьной строки или количество элементов в массиве. ltrim( <cString> ) --> cTrimString Удаляет начальные пробелы из символьной строки. memoread( <cFile> ) --> cString Возвращает содержимое дискового файла как строку символов. memowrit( <cFile>, <cString> ) --> lSuccess Записывает символьную строку cStr в файл с именем cFile. padc( <exp>, <nLength>, [<cFillChar>] ) --> cPaddedString Превращает символьное или числовое выражение ехр в строку символов длиной nLenth и размещает результат в центре, добавляя символ cFillChar до получения длины nLength. padl( <exp>, <nLength>, [<cFillChar>] ) --> cString Превращает символьное или числовое выражение ехр в строку символов длиной nLength и добавляет к левой стороне строки

символ cFillChar до получения длины nLength.

padr( <exp>, <nLength>, [<cFillChar>] ) --> cString Превращает символьное или числовое выражение ехр в строку символов длиной nLength и добавляет к правой стороне строки символ cFillChar до получения длины nLength. rat( <cSearch>, <cTarget> ) --> nPosition Возвращает позиции последнего вхождения (справа) подстроки cSearch в строку cTarget. replicate( <cString>, <nCount> ) --> cRepeatedString Возвращает строку, повторенную заданное количество раз. rtrim( <cString> ) --> cTrimString Удаляет конечные пробелы из символьной строки. space( <nCount> ) --> cSpaces Возвращает строку из nCount пробелов. str( <nNumber>, [<nLength>], [<nDecimals>] ) --> cNumber Преобразует число nNumber в строку символов длиной nLength с nDecimals цифрами после десятичной точки. strtran(<cString>,<cSearch>,[<cReplace>],[<nStart>],[<nCount>]) --> cNewString Ищет в строке cString включение строки cSearch и заменяет cSearch на cReplace. Примеры: cString := "Вычислять или не вычислять ?" cString := StrTran( cString, "Вычислять", "Быть" ) = = "Быть или не быть ?" stuff( <cString>, <nStart>, <nDelete>, <cInsert> ) --> cNewString Удаляет или вставляет символы в строке. Примеры: stuff("Коза в огороде",1,4,"Козел") = "Козел в огороде" stuff("Коза в огороде",5,0," и козел") = "Коза и козел в огороде" substr( <cString>, <nStart>, [<nCount>] ) --> cSubString Возвращает часть строки cString, начиная с символа с номером nStart и длиной nCount. Примеры: substr("Коза в огороде",1,4) = "Коза" substr("Kosa b oropoge",5) = " b oropoge" val( cNumber ) --> nNumber Преобразует число в символьной форме в числовой тип. cString1 \$ cString2 --> lSuccess Возвращает признак вхождения строки cString1 в строку cString2. Если строка cString1 встречается в строке cString2, то возвращается .t. SaveScreen() - сохранить часть экрана в строке символов. Пример: s:=SaveScreen( 2,4,5,62 ) сохраняет в переменной в часть экрана между строками 2 и 5 и между колонками 4 и 62. Для восстановления используется функция RestScreen() RestScreen() - восстанавливает часть экрана из ранее сохраненной

```
строки символов.
         Пример:
         RestScreen( 2,4,5,62,s )
         восстанавливает из переменной в часть экрана между строками
         2 и 5 и между колонками 4 и 62.
         Для сохранения экрана используется функция SaveScreen().
   SetCursor() - запоминает/восстанавливает характеристики курсора.
         Пример:
         n:=SetCursor()
            . . .
         SetCursor(n)
   SaveREC() - запоминает положение в электронном каталоге.
  RestREC() - восстанавливает положение в электронном каталоге.
   Setpos() - устанавливает курсор в заданную позицию.
        Пример:
         setpos(4,6) - устанавливает курсор на 4 строку 6 колонки.
   Qout(), Qqout() - выдает информацию на экран. Qout() - выдает
         информацию с новой строки, Qqout() - с текущей позиции.
Приложения
     Список меток по каталогам
     Список меток, заносимых в поисковые таблицы
    Коды языков
    Коды клавиш
   Альтернативная кодировка
```

## Список меток по каталогам

КНГ - Каталог Книги КМП - Каталог Комплектование СТТ - Каталог Статьи ПРД - Каталог Периодика ПДП - Каталог Подписка

Метка	Название Д	(лина	Н	азван	ие ка	талог	ОВ	
1008	Тип ФИО 0,1,2,3	1	КНГ	КМП	CTT			
100a	Первый автор ФИО	50	КНГ	КМП	CTT			
100b	Династ.номер	16	КНГ	КМП	CTT			
100c	Титул (зван.)	24	КНГ	КМП	CTT			
100d	Лата	16	КНГ	КМП	CTT			
100e	Роль лин	2.4	кнг	КМЦ	СТТ			
7008	Тип ФИО 0.1.2.3	1	кнг	КМЦ	СТТ			
700a		48	кнг	кмп	CTT			
700e	Ропь пин	24	кнг	КМЦ	CTT			
1108		1	кнг	КМЦ	CTT	прп	ΠΠΠ	
110a		180	кнь	КМЦ	CTT	прп	ппп	
110c	Место	32	кнг	КМЦ	CTT	прп	ппп	
110C		120	VUT VUT	LIMIT		прп	ппп	
110D	подчин.звено	16	KUL	ICMIT		прп	плп	
	номер	10	NUL			прд	пдп	
110a	дача	100	KHI.	KMII	CTT	прд	пдп	
1110 1110	проч.информ.	10U	KHI'	KMII KMII	CTT	прд	11Д11 н н н	
1118	порядок 0,1	⊥ 100	КНГ	КМП	CTT	прд	11Д11	
⊥⊥⊥a	врем.кол.автор	180 180	КНГ	КМП	CTT	прд	11ДП	
111e	Подчин. звено	120	КНГ	КМП	CTT	ПРД	ПДП	
111n	Номер	20	КНГ	КМП	CTT	ПРД	ПДП	
111d	Дата	20	КНГ	КМП	CTT	ПРД	ПДП	
111c	Место	40	КНГ	КМП	CTT	ПРД	ПДП	
111g	Прочая информ.	100	КНГ			ПРД	ПДП	
2458	Продолжение 0,1	1	КНГ	КМП	CTT	ПРД	ПДП	
2459	Незн.симв. 0-9	1	КНГ	КМП	CTT	ПРД	ПДП	
245k	Форма	20				ПРД	ПДП	
245a	Заглавие	256	КНГ	КМП	CTT	ПРД	ПДП	
245n	Номер части	16	КНГ	КМП		ПРД	пдп	
245p	Название части	120	КНГ	КМП		ПРД	пдп	
245b	Продолж.загл.	120	КНГ	КМП	CTT	ПРД	ПДП	
245c	Свед.об ответ.	180	КНГ	КМП	CTT	ПРД	ПДП	
2468	Добавочная -0,1	1				ПРД	пдп	
2469	Тип заглавия 1	1				ПРЛ	плп	
246a	Лругие заглавия	240				прл	плп	
246f		10				прп	ΠΠΠ	
2428	Побавочн -0 1	1	кнг	кмп	CTT	прп	ΠΠΠ	
2120	HODE CAMP $-0-9$	1	КПГ	КМЦ	CTT	прп	ппп	
2727 2125		⊥ 255	KIII VUT	K MII	CTT	прп	ппп	
272a 94977	Перевод заглавия	100	1/111 1/111	1/14111	CII	ше д	*14411	
272y 2170	лобороннос О 1	1	VUT.			прп	ΠΠΠ	
24/0 0/7-	дооавочная-О,1	⊥ 70				шед пра	пдп	
24/a	пред. заглавие	70				ШРД ПРЛ	11Д11	
∠4/1	номер/год/вып.	50				прд	11Д11	
∠⊥48	доравочная-0,1	1				прд	11Д11	
∠⊥4a	название для РП	80				ΠРД	11Д11 — — —	
022a	Индекс ISSN	16				ПРД	пдп	
4909	Незн.симв. 0-9	1	КНГ	КМП		ПРД	ПДП	
490a	Серия	120	КНГ	КМП		ПРД	ПДП	
490x	ISSN серии	10	КНГ	КМП		ПРД	пдп	
490v	Номер серии	16	КНГ	КМП		ПРД	ПДП	
310a	Периодичность	80				ПРД	ПДП	
3628	Формат данных	1				ПРД	ПДП	
362a	Спецификация	100				ПРД	пдп	
362z	Источник данных	100				ПРД	пдп	
043a	Страна	7				ПРД	ПДП	
0 - 0	- -	2.2		LINT				

250b	Свед.отн. к изд.	120	КНГ	КМП			
2608	Издат.? 0,1	1	КНГ	КМП		ПРД	ПДП
260a	Место издания	60	КНГ	КМП		ПРД	ПДП
260b	Издательство	120	КНГ	КМП		ПРД	ПДП
260c	Дата издания	16	КНГ	КМП			
773a	Загол.источника	120			CTT		
773t	Загл.источника	120			CTT		
773b	Свед. об изд.	40			CTT		
773d	Выходные данные	40			CTT		
773g	Смешан.информ.	120			CTT		
773k	Серия	40			CTT		
773x	ISSN источника	9			CTT		
773z	ISBN источника	16			CTT		
300a	Объем	16	КНГ	КМП	CTT		
300b	Иллюстрации	32	КНГ	КМП	CTT		
300c	Размер	16	КНГ	КМП		ПРД	ПДП
350a	Цена комплекта	40				ПРД	пдп
515a	Обл. нумерации	999				ПРД	плп
550a	Органы НТИ ?	1				ПРД	плп
500a	Примечания	240	КНГ	КМП	CTT	ПРД	плп
520a	Реферат	999	КНГ	КМП	CTT	ПРД	плп
650a	Предмет, рубрика	120	КНГ	КМП	СТТ	прл	плп
650x	Полрубрика	120	КНГ	КМП	CTT	прл	плп
653a	Ключевые слова	120	кнг	КМЦ	СТТ	прл	плп
020a	Инлекс ISBN	16	кнг	КМЦ	011		
020c	Пена	16	кнг	КМЦ			
093a	Номер в ТП	20		КМЦ			
093b	Номер КБ	20		КМЦ			
583a	Лействие	20		КМЦ			
583c	Лата	9		КМП			
583b	Ситла	16		КМЦ			
583n	Количество	4		кмп			
583k	Размешение з-за	- 50		КМЦ			
583d	Контрольный срок	9		кмп			
583x	Номера счетов	20		КМЦ			
080a	Νηπεκς ΥΠΚ	24	кнг	КМЦ	СТТ		
091a	Инлекс ББК	2.4	кнг	КМЦ	СТТ		
092a	Инлекс ГАСНТИ	2.4	кнг	КМЦ	СТТ	прл	плп
090c	Каталожные инл.	2.4	кнг	КМЦ	СТТ	прл	плп
032a	Инлекс РП	10			011	прл	плп
852b	Сигла	12	кнг	кмп	СТТ	прл	плп
852t	Копичество экз	32	кнг	кмп	СТТ	прл	плп
852h	Попочный инлекс	16	кнг	кмп	СТТ	прл	плп
852i	Авторский знак	8	кнг	кмп	СТТ	прл	плп
852p	Инвент номер	240	кнг	кмп	011		******
863+	Коп экземппара	10	КНЦ	1 (1 11 1			
040a	Название оплан	70	КНЦ				
041a	Язык текста	12	КНЦ	кмп	Стт	прп	плп
0087	KOUND DARANG	40	КНЦ	КМЦ	CTT CTT	трп	ппп
5007	тодяр и данные	10	T/TTT	1/1 111	<u> </u>	*** 44	*****

## Каталог Читатели

Метка	Название	Длина
920n	Номер ч/б	20
920a	ФИО читателя	120
920b	Образование	40
920c	Категория	40
920d	Домашний адрес	250
920e	Телефон (дом.)	20
920f	Данные паспорта	120
921a	Место раб./Учеб.	120
921b	Отделение	120

921c	Кафедра/Подразд.	120
921d	Курс/Год обуч.	20
921e	Группа	20
921f	Телефон (раб.)	20
922a	Место регистр.	120
922b	Дата регистр.	20
922c	Перерегистрация	20
923b	Код книги	10

Каталог Книги\_МГУ

Метка	Название	Длина
1008	Индик.типа имени	1
100a	Первый автор ФИО	40
100b	Нумерация	6
100c	Титул (звание)	24
100q	Полная форма им.	40
100d	Даты	16
100e	Родственный терм	40
100u	Разное	24
100t	Название работы	120
100k	Форма подзагол.	32
100n	Номер части	б
100p	Название части	120
7008	Индик.типа имени	1
700a	Другие авт.ФИО	50
700b	Нумерация	6
700c	Титул (звание)	24
700q	Полная форма им.	40
700a	Даты	10
700e	Родств. термин	40
700u 700+		24 120
7000	пазвание рассты	220
700k 700n		52
700m	Название насти	120
1108		120
110a		60
110b	Полчинен звено	60
110n	Порялковый номер	6
110d	Лата	10
110c	Место	24
110g	Прочая информ.	60
7108	Порядок - 0,1,2	1
710a	Др.пост.кол.авт.	60
710b	Подчинен.звено	60
710n	Порядковый номер	б
710d	Дата	10
710c	Место	24
710g	Прочая информ.	60
1118	Порядок - 0,1,2	1
111a	Врем.кол.автор	60
111e	Подчинен. звено	60
111n	Порядковый номер	6
111d	Дата	10
111c	Место	24
111g	Прочая информ.	60
7118	Порядок - 0,1,2 -	1 C O
711a	др.врем.кол.авт.	6U
/11e	подчинен. звено	6U
/11n	порядковыи номер	6 1 0
711d	Дата	ΤU

24 711c Место 711g Прочая информ. 60 2458 Индик.доб.опис. 1 2459 Незн.симв. 0-9 1 240 245а Заглавие 200 245b Продолж.загл. 10 245n Номер части 245рНазвание части120245сСвед.об ответ.120 2428 Индик.доб.описан 1 2429 Незнач.симв.0-9 1 242а Перевод заглавия 240 242b Продолж. заглав. 60 242с Свед.об ответ. 60 б 242n Номер части 242р Название части 120 242у Язык перев.загл. 3 250а Свед.об издан. 32 250b Свед.отн. к изд. 60 2608 Свед.об издат. 1 260а Место издания 60 260b Издательство 80 260с Дата издания 16 
 300а
 Объем
 16

 300b
 Иллюстрации
 50

 300c
 Размеры
 16
 300е Сопровод.матер. 30 4908 Индик.трассиров. 1 4909 Незнач.символы 1 490а Заглавие серии 120 490v Homep Toma 6 490x ISSN 10 500а Общие примечания 240 500z Источ.информации 50 501а Прим.о содерж. 120 502а Прим. о дисс. 200 504а Прим.о справ.ап. 200 
 504a
 Прим.о
 20

 504b
 Кол-во ссылок
 20

 7
 20
 999
 8668 Инд.уровня опис. 1 8669 Инд.типа описан. 1 8666 Поле связи 
 8666
 Поле связи
 2

 866a
 Инф.о хранении
 10

 866z
 Примечания
 50

 650a
 Предмет.рубрика
 120

 650x
 Подрубрика
 120

 653a
 Ключевые слова
 60

 020a
 Индекс ISBN
 16

 020c
 Цена
 16

 080a
 Индекс УДК
 24

 091a
 Индекс ББК
 24

 092a
 Индекс ГАСНТИ
 24

 090c
 Каталожные инд.
 30

 090g
 Номер путевки
 10
 2 090д Номер путевки 10 852b Сигла 20 
 852t
 Количество экз.
 32

 852h
 Полочный индекс
 16

 852i
 Авторский знак
 10
 852р Инвент.номер 20 040b Язык каталогиз. 6 0418 Признак.перев. 1 041а Язык текста 12 041b Язык аннотации 12 041h Язык оригинала 10

# 0087 Кодир. данные 40

!Длина поля – понятие довольно условное, т.к., во-первых, длину всегда можно увеличить, а во-вторых, используя в редакторе Marc-Edit клавишу F3, можно вводить данные большей длины, чем установлено. При таком вводе ограничений на длину нет. Список меток, заносимых в поисковые таблицы

## Каталог Книги

Элемент	Дл.инд.	Дл.поля	Метки
автору	20	20	100a 700a
заглавию	10	50	245a
коллек.автору	10	30	110a 710a 111a 711a 245c
названию серии	10	30	490a
предмет.рубрика	м 10	25	650a 650x
ключевым словам	10	25	653a
индексам ГАСНТИ	8	8	092a
индексам УДК	16	16	080a
индексам ББК	16	16	091a
индексам ISBN	11	11	020a
инвент.номеру	10	10	TIN
месту хранения	8	8	852b
издательству	10	10	260b
году издания	4	4	260c
коду экземпляра	10	10	863t
языку	10	10	041a
типу литературы	8	8	952a
номеру КСУ	10	10	TSU

!Если метка начинается с символа t, то это по сути не метка, а имя файла, в котором хранятся правила, по которым будет обрабатываться информация перед занесением в поисковую таблицу. Так, если метка tin, то это означает, что правила занесения хранятся в файле tin.blc. После t могут быть только два любых символа.

## Каталог Комплектование

Элемент	Дл.инд.	Дл.поля	Метки
автору	32	32	100a 700a
заглавию	50	50	245a
коллек.автору	30	30	110a 710a 111a 711a
названию серии	30	30	440a 440p
номеру в ТП	20	20	093a
номеру КБ	20	20	093b
сигле заказа	10	10	583b
номеру счета	10	10	583x
индексам ISBN	11	11	020a
дате заказа	10	10	583c
издательству	10	10	260b
году издания	4	4	260c
контрольн. сроку	10	10	583d

### Каталог Статьи

Элемент	Дл.инд.	Дл.поля	Метки
автору	20	20	100a 700a
заглавию	50	50	245a
коллек.автору	30	30	110a 710a 111a 711a 245c
источнику	30	30	773a 773t
предмет.рубрика	ам 25	25	650a 650x
ключевым словам	a 25	25	653a
индексам УДК	16	16	080a
индексам ББК	16	16	091a

индексам ISBN	11	11	020a
месту хранения	8	8	852b

## Каталог Периодика

Элемент	Дл.инд.	Дл.поля	Метки
коллек.автору	30	30	110a 110g 111a 111g
заглавию	50	50	245a 210a 242a 246a 247a
названию серии	30	30	490a
индексам РП	10	10	032a
виду издания	10	10	245k
индексам ISSN	11	11	022a
сигле хранения	8	8	852b
стране	7	7	043a
ключевым словам	4 1O	25	653a

## Каталог Подписка

Элемент	Дл.инд.	Дл.поля	Метки
заглавию	10	50	245c
индексам РП ситле	10	10	032a 852b
форме	10	10	245k
стране	7	7	043a
индексам ISSN	11	11	022a
каталогу	20	20	550a

## Каталог Читатели

Элемент	Дл.инд.	Дл.поля	Метки			
номеру ч/б	10	10	920n			
ФИО читателя	10	30	920a			
месту работы	10	30	921a	921b	921c	
месту регистр.	10	30	922a			
месту выдачи	10	10	923a			
коду книги	10	10	923b			
дате регистрац	. 8	8	922b			

# Каталог Книги\_МГУ

Элемент	Дл.инд.	Дл.поля	Метки	
				_
автору	16	16	100a 700a 100q 700q	
названию	32	32	tti	
кол.автору	32	32	110a 710a 111a 711a	
предм.рубрикам	24	24	650a 650b 650y 650z 650	x
ключевым словам	4 24	24	653a 650a 650x 650y 650	z
индексам УДК	16	16	080a	
индексам ББК	16	16	091a	
индексам ISBN	10	10	020a	
индексам ГАСНТИ	1 10	10	092a	
сиглам хранения	н 8	8	852b 090f	
индексам НБ МГУ	/ 16	16	tca	
инв.номерам	16	16	tin	
издательству	32	32	260b	

серии	32	32	tse
году издания	4	4	260c
месту издания	16	16	260a
парал.заглавию	30	30	242a
полочному инд.	10	10	852h 852i

Коды языков

азербайджанский aze английский eng арабский ara армянский arm bel белорусский bul болгарский hun венгерский vie вьетнамский cre греческий geo грузинский dan датский **spa** испанский ita итальянский kaz казахский kir киргизский chi китайский kor корейский lav латышский lit литовский mol молдавский mon монгольский ger немецкий польский pol португальский por румынский rum русский rus slo словацкий taj таджикский tuk туркменский uzb узбекский ukr украинский fre французский hin хинди чешский cze шведский swe эстонский est японский jpn

Название языка

Код

Коды клавиш

F1	28	Shift+F1	-10	Ctrl+F1	-20	Alt+F1	-30
F2	-1	Shift+F2	-11	Ctrl+F2	-21	Alt+F2	-31
F3	-2	Shift+F3	-12	Ctrl+F3	-22	Alt+F3	-32
F4	-3	Shift+F4	-13	Ctrl+F4	-23	Alt+F4	-33
F5	-4	Shift+F5	-14	Ctrl+F5	-24	Alt+F5	-34
Fб	-5	Shift+F6	-15	Ctrl+F6	-25	Alt+F6	-35
F7	-6	Shift+F7	-16	Ctrl+F7	-26	Alt+F7	-36
F8	-7	Shift+F8	-17	Ctrl+F8	-27	Alt+F8	-37
F9	-8	Shift+F9	-18	Ctrl+F9	-28	Alt+F9	-38
F10	-9	Shift+F10	-19	Ctrl+F10	-29	Alt+F10	-39

Альтернативная кодировка

Коды с 0 по 127 в десятичной системе

Код	Сим-	Код	Сим-	Код	Сим-	Код	Сим-1	Код	Сим-	Код	Сим-	Код	Сим-	Код	Сим-
	вол		вол		вол		вол		вол		вол		вол		вол
000		016		032		048	0	064	1 @	080	Ρ	096	•	112	р
001		017		033	!	049	1	065	5 A	081	Q	097	а	113	q
002		018		034	"	050	2	066	5 В	082	R	098	b	114	r
003		019		035	#	051	3	067	7 C	083	S	099	С	115	S
004		020		036	\$	052	4	068	3 D	084	Т	100	d	116	t
005		021		037	00	053	5	069	) E	085	U	101	е	117	u
006		022		038	&	054	б	070	) F	086	V	102	f	118	v
007		023		039	1	055	7	071	L G	087	W	103	g	119	W
800		024		040	(	056	8	072	2 Н	088	Х	104	h	120	x
009		025		041	)	057	9	073	3 I	089	Y	105	i	121	У
010		026		042	*	058	:	074	ł J	090	Z	106	j	122	z
011		027		043	+	059	;	075	5 К	091	[	107	k	123	{
012		028		044	,	060	<	076	5 Г	092	$\backslash$	108	1	124	
013		029		045	-	061	=	077	7 M	093	]	109	m	125	j
014		030		046		062	>	078	8 N	094	^	110	n	126	~
015		031		047	/	063	?	079	) ()	095		111	0	127	

Коды с 128 по 255 в десятичной системе

Код	Сим-	Код	Сим-	Код	Сим-	Код	Сим-	Код	Сим-	Код	Сим-	Код	Сим-	Код	Сим-
	вол		вол		вол		вол		вол		вол		вол		вол
128	A	144	Ρ	160	а	176		192	<u>i</u> L	208	Ш	224	р	240	Ë
129	Б	145	С	161	б	177		193	3 L	209	╤	225	С	241	ë
130	В	146	Т	162	в	178		194	ιτ	210	π	226	т	242	E
131	Г	147	У	163	г	179	T	195	;	211	Ш.	227	У	243	e
132	Д	148	Φ	164	д	180	-	196	5 —	212	F	228	Ф	244	Ï
133	Ε	149	Х	165	е	181	=	197	' +	213	F	229	Х	245	ï
134	Ж	150	Ц	166	ж	182	-1	198	3  =	214	Г	230	Ц	246	Ў
135	З	151	Ч	167	з	183	л П	199	) <u></u> }	215	₽	231	ч	247	ў
136	Ν	152	Ш	168	И	184	Ę	200	) Ľ	216	ŧ	232	ш	248	0
137	Й	153	Щ	169	й	185	f	201	- F	217	L	233	щ	249	•
138	К	154	Ъ	170	к	186		202	<u>1</u>	218	г	234	ъ	250	•
139	Л	155	Ы	171	л	187		203	3 1	219		235	ы	251	$\checkmark$
140	М	156	Ь	172	М	188	Ŀ	204	⊧ ⊫	220		236	Ь	252	N⁰
141	Н	157	Э	173	Н	189	Ш	205	5 =	221		237	Э	253	¤
142	0	158	Ю	174	0	190	Ę	206	5 #	222		238	Ю	254	
143	П	159	Я	175	П	191	٦	207	, <u>⊥</u>	223		239	Я	255	