### Общие сведения

Приложение Web-ИРБИС64+PHP представляет собой решение для эксплуатации различных поколений интерфейсов Web-ИРБИС64 на ОС Windows и ОС Linux. Само приложение написано на PHP, а для взаимодействия с базами данных САБ ИРБИС64 используется модуль САБ ИРБИС64 для платформы PHP. Модуль САБ ИРБИС64 для PHP поставляется для различных версий платформы PHP для ОС Windows и для ОС Linux. От обычного модуля Web-ИРБИС64 приложение отличается более высокой скоростью работы, расширенными возможностями кастомизации как интерфейсной части, так и процесса взаимодействия с другими ресурсами библиотеки, а также наличием Linux версий.

### Системные требования

PHP версии 5.6.0 и выше с установленными модулями gd2, mbstring.

### Варианты поставки

Для Windows

1) Установщик с набором модулей для нескольких актуальных версий PHP.

2) Xampp-portable

Приложение предоставляется вместе с веб сервером xampp и актуальной интерфейсной частью 2019+. В этом случае потребуется минимальная настройка для связи с базами данных САБ ИРБИС64.

3) Модуль для конкретной версии php (например для работы через FastCGI с веб серверами nginx или IIS). В этом случае потребуется самостоятельно подключить модуль САБ ИРБИС64 к дистрибутиву имеющимуся php и настроить интерфейсную часть.

Для Linux

Модуль для конкретной версии php в конкретной версии операционной системы. В этом случае потребуется самостоятельно подключить модуль САБ ИРБИС64 к дистрибутиву php и настроить интерфейсную часть.

### Общие сведения о настройке приложения

Если приложение было получено вместе с веб сервером и PHP, либо в виде модуля САБ ИРБИС64 для PHP и интерфейсной части, то для начала работы потребуется откорректировать пути к PHP и к дистрибутиву САБ ИРБИС64 файлах xampp/htdocs/web/config.inc и xampp/htdocs/web/irbis\_server\_ft\_php.ini.

**Файл config.inc**

//путь к интерпретатору PHP

define('PHP\_PATH', 'E:/xampp/php/php.exe');

//путь к ини файлу интерфейсной части

define('INI\_PATH', 'E:/xampp/htdocs/web/irbis\_server\_ft\_php.ini');

//писать или нет в LogDB

define('LOG\_ENABLED', FALSE);

**Файл irbis\_server\_ft\_php.ini**

Секция MAIN

[MAIN]

IRBISPATH=C:\Irbis64\_18plus\IRBIS64\

DATAPATH=C:\Irbis64\_18plus\IRBIS64\DATAI\

DepositPATH=E:\xampp\htdocs\web\deposit\

workdir=C:\Irbis64\_18plus\IRBIS64\workdir

ACTABPATH=C:\Irbis64\_18plus\IRBIS64\isisacw

UCTABPATH=C:\Irbis64\_18plus\IRBIS64\isisucw

SYSPATH=C:\Irbis64\_18plus\IRBIS64\

CgiFlc=E:\xampp\htdocs\web\deposit\cgiflc.pft

LIBDIR=E:\xampp\php\ext\

Кроме этого, в секции MAIN требуется указать актуальный лицензионный ключ

LIB\_NAME=Тестовая библиотека

LICENSE\_KEY=eaaaa4aaaeb8d7c7-L44556643c4e4bndc1-17f0oooo58830a3e

Секции, относящиеся к конкретным базам данных

Потребуется откорректировать пути к папкам интерфейсной части.

Пример.

[IBIS]

FRAMES=E:\xampp\htdocs\web\frames\_r\FullText\

Более подробную информацию см. в разделе **Настройка приложения Web-ИРБИС64+PHP и интерфейсной части Web-ИРБИС64.**

### Использование приложения Web-ИРБИС64+PHP с модулем САБ ИРБИС64, предоставленным для вашей версии PHP

Процесс развертывания приложения можно разделить на следующие этапы.

1. Получение версии модуля САБ ИРБИС64 для вашей версии PHP
2. Подключение модуля САБ ИРБИС64 к PHP
3. Настройка приложения Web-ИРБИС64+PHP и интерфейсной части Web-ИРБИС64
4. При использовании интерфейсной части Web-ИРБИС64 с собственными модификациями потребуется выполнить поиск и устранение ошибок в форматах. Это связано с тем, что в модуле САБ ИРБИС64 для PHP используется интерпретатор языка форматирования xpft, который ведет себя более строго в отношении ошибок синтаксиса. Последние версии интерфейсной части Web-ИРБИС64 поколений 2017, 2018 и 2019 уже проверены на соответствие требованиям xpft, однако более ранние версии, которые могут быть в наличии у пользователей потребуется проверить.

### Получение модуля САБ ИРБИС64 для PHP

Для тех пользователей, у которых на сервере нет установленной версии PHP, доступен вариант поставки модуля САБ ИРБИС64 для PHP вместе с PHP и веб сервером Apache в составе пакета Xampp-portable (описано выше). В этом случае потребуется только отредактировать ini файл приложения (обычно irbis\_server\_ft\_php.ini, но можно назначить любой путь) и файл config.inc.

Если веб сервер и PHP на сервере уже присутствуют и требуется подключить модуль САБ ИРБИС64 к ним, тогда понадобится, прежде всего, узнать версию PHP. Когда номер версии PHP получен, эту информацию следует указать при заявке на поставку приложения.

**Для пользователей Windows.**

Найдите место на диске, где находится приложение php.exe, откройте командную строку, переместитесь в это место и запустите в ней php.exe с ключом –v.

Например

Переход в место, где находится php.exe

D:\>cd d:\php-7.0.6-Win32-VC14-x64

Вывод версии (номер версии выделен жирным)

D:\php-7.0.6-Win32-VC14-x64>php -v

**PHP 7.0.6 (cli) (built: Apr 28 2016 13:46:50) ( ZTS )**

Copyright (c) 1997-2016 The PHP Group

Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies

**Для пользователей Linux.**

Как правило, на сервере под управлением ОС Linux PHP доступен в составе дистрибутива. Однако, возможны варианты, когда PHP может быть собран самостоятельно или установлен из альтернативного репозитория. Для получения модуля PHP для вашей ОС потребуется получить информацию о версии операционной системы (например, Ubuntu 18.04). Кроме этого, когда PHP получен не из состава дистрибутива, потребуется эту информацию также указать.

Чтобы получить номер версии PHP в Linux требуется выполнить в командной строке команду php -v.

Например (номер версии выделен жирным).

$ php -v

**PHP 5.6.40 (cli) (built: Dec 17 2019 15:01:55)**

Copyright (c) 1997-2016 The PHP Group

Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies

### Подключение модуля САБ ИРБИС64 к PHP

Для подключения модуля САБ ИРБИС64 к установленной версии PHP потребуется скопировать файл модуля и сопутствующие dll в директорию, где находятся другие модули расширений и указать в ini файле PHP, что указанный модуль требуется подключать при запуске.

**Для пользователей Windows.**

Обычно модули PHP находятся в папке ext. Например, если PHP расположен по пути D:\php-7.0.6-Win32-VC14-x64, то модули находятся в D:\php-7.0.6-Win32-VC14-x64\ext. Файл модуля САБ ИРБИС64 требуется скопировать туда. Если PHP получен путем копирования папки с файлами и не настроен (например, скачан с сайта php.net), требуется выполнить несколько действий:

1. Если файла php.ini в каталоге PHP нет, требуется скопировать один из файлов – примеров php.ini-development или php.ini-production, скопированный файл переименовать в php.ini
2. В файле php.ini найти параметр extension\_dir. Если он закомментирован (стоит точка с запятой в начале), то указать значение, соответствующее текущему пути, например extension\_dir = "d:/php-7.0.6-Win32-VC14-x64/ext"

Далее требуется добавить строчку с именем модуля САБ ИРБИС64 для автоматической загрузки.

Для PHP 5 и ранних версий PHP 7 требуется указывать имя модуля с префиксом php\_ (можно посмотреть есть ли префикс php\_ на других строчках с именами модулей)

Пример

extension=php\_irbis64.dll

Для актуальных версий PHP 7 (7.2 и выше) в имя модуля уже не включается префикс php\_ (можно это понять по другим строчкам с именами модулей)

extension=irbis64.dll

**Для пользователей Linux.**

Обычно файлы PHP разбросаны по нескольким разным местам в системе. В архиве с модулем САБ ИРБИС64 находятся файлы с теми путями, как они расположены в конкретной версии операционной системы. Как правило, в каталоге /etc находятся файлы с настройками, в /usr бинарные файлы, в /var или /srv – файлы веб сервера.

Например, для Ubuntu 18.04

Содержимое каталога distrib/etc требуется скопировать в /etc.

Содержимое каталога distrib/usr в /usr.

Содержимое каталога distrib/var в /var.

**Проверка загрузки модуля САБ ИРБИС64 в PHP**

После того, как модуль САБ ИРБИС64 подключен к PHP, требуется выполнить скрипт

<?php

phpinfo();

?>

В выводе данного скрипта должна присутствовать информация о модуле IRBIS64.

### Настройка приложения Web-ИРБИС64+PHP и интерфейсной части Web-ИРБИС64

После того, как модуль САБ ИРБИС64 был подключен к вашему дистрибутиву PHP, нужно выполнить несколько действий по настройке приложения. В случае, когда приложение получено вместе с веб сервером Xampp-portable, все файлы приложения находятся в директории htdocs/web.

**Настройка файла config.inc**

Нужно, как минимум, указать путь к ini файлу интерфейсной части Web-ИРБИС64 и путь к интерпретатору PHP для полнотекстовых команд.

Найдите строку

define('PHP\_PATH', 'E:/xampp/php/php.exe');

Укажите в ней действующий путь к php.exe.

Найдите строку

define('INI\_PATH', 'E:/xampp/htdocs/web/irbis\_server\_ft\_php.ini');

Укажите в ней путь к ini файлу интерфейсной части Web-ИРБИС64

В том случае, если используется классический интерфейс 2017 (и более ранний) с базами 2018+ и более поздних версий, требуется указать TRUE вместо FALSE в следующей строке

define('COMPAT\_2017', FALSE);

Также может понадобиться сменить параметры по умолчанию для стартовой страницы

define('DEFAULT\_GET\_PARAMS', '?LNG=&C21COM=F&I21DBN=IBIS&P21DBN=IBIS');

Если нужна запись информации в базу данных LogDb

define('LOG\_ENABLED', TRUE);

**Настройка файла irbis\_server\_ft\_php.ini**

В качестве ini файла интерфейсной части может использоваться обычный ini файл Web-ИРБИС64. Имя файла, также может быть любым, оно определяется в config.inc.

В случае, если используется ini файл Web-ИРБИС64, который не был получен вместе с дистрибутивом Web-ИРБИС64+PHP, для корректной работы PHP модуля понадобится добавить несколько опций в секцию MAIN.

Путь к бинарным файлам модуля (дополнительные dll для работы с pdf файлами), можно скопировать их в директорию с расширениями PHP и указать ее.

LIBDIR=E:/xampp/php/ext/

Информация о лицензии (имя и ключ)

LIB\_NAME=Тестовая библиотека

LICENSE\_KEY=e45554af1118d7c7-L6yyyd211c4ec6dc1-1xxx100758830a3e

Для работы приложения требуется наличие лицензионного ключа (параметр LICENSE\_KEY в секции MAIN). При наличии некорректного ключа приложение переходит в демонстрационный режим, при этом часть функций будут недоступны.

Также, при настройке ini файла следует убедиться в корректности путей в параметрах.

**Настройка относительных url приложения в irbis\_server.ini**

Параметры секции [WEB] определяют url, по которым будет доступно настроенное приложение.

Значения по умолчанию.

HTTP\_PATH=/web/irbis64r

CGI\_PATH=/web

CGI\_ALIAS=/web/index.php

Настройки по умолчанию подразумевают, что приложение будет доступно по любому доступному для веб сервера адресу с относительным путем /web

Например: <http://localhost/web/>

В случае, если требуется перенести приложение в другое место, например в корень веб сервера, то, помимо корректирования config.inc и путей в irbis\_server\_php.ini, потребуется отредактировать эти 3 параметра.

Финальная настройка для пользователей ОС Linux

В том случае, если приложение Web-ИРБИС64+ PHP используется вместе с другими, приложениями, использующими базы данных ИРБИС, потребуется все приложения ИРБИС настроить на работу с одной и той же учетной записью в операционной системе. Это связано с тем, что для синхронизации доступа на запись к файлам баз данных используются системные объекты, размещенные в разделяемых между процессами сегментах памяти (shared memory segments). В ОС Linux данные сегменты памяти находятся в виртуальной файловой системе /dev/shm и выглядят как обычные файлы. Проблема состоит в том, что, несмотря на то, что система предоставляет доступ к сегментам памяти как к обычным файлам, ядро системы игнорирует настройку прав доступа к ним для других пользователей. Это приводит к тому, что процессы, запущенные под одним пользователем не смогут открыть на запись объект синхронизации для доступа к базе данных, если его уже создал другой процесс под другим пользователем. Из-за этого запись данных в базу может оказаться недоступной. В том случае, если такая ситуация произошла, нужно убедиться, что в данный момент не производится запись в базу и удалить из /dev/shm файлы с именем базы данных в названии, например IRBIS\_MUTEX\_RDR. Также можно перезагрузить операционную систему. Файлы в /dev/shm после перезагрузки исчезнут.

### Изменения относительно CGI приложения Web-IRBIS64 и Web-IRBIS64+

Для полнотекстовых баз данных не требуется хранение координат слов в базе, что существенно ускоряет процесс индексирования. Индексирование слов в ПБД определяется параметром ExtractPDFLink в секции FullText, данный параметр можно установить в 0. Вместо этого, координаты слов текстового слоя сохраняются в каталоге cache, который создается автоматически в каталоге, определяемом параметром CGIPATH в секции WEB.

В команде обновления данных R добавлен необязательный параметр R21AUTOIN, который означает включение автоввода при сохранении записи и может присутствовать в каждой группе полей. По умолчанию автоввод включен.

### Для разработчиков

При использовании интерфейсной части Web-ИРБИС64 с модулем Web-ИРБИС64+PHP появились дополнительные возможности для разработчиков.

1. Использование функционала САБ ИРБИС64 непосредственно в языке PHP аналогично использованию irbis64.dll в приложениях
2. Использование функций PHP в языке форматирования, в том числе перехватчик функций unifor
3. Использование тегов PHP внутри фреймов \*.frm, вызов кода PHP при вторичной обработке вывода форматов.
4. Отладка ошибок в форматах с помощью вывода в поток ошибок PHP и в файл xpft\_error.log
5. Тег <?xml ?> выводится без обработки (для формирования xml документов)

### Использование функций PHP в языке форматирования

Использование функций PHP в языке форматирования ИРБИС доступно для обоих интерпретаторов: классического и xpft. Переключение интерпретаторов языка форматирования осуществляется функцией irbis\_enable\_xpft. По умолчанию включен классический интерпретатор. Чтобы использовать функции, написанные на PHP в скриптах на языке форматирования ИРБИС, потребуется их оформить определенным образом и указать в списке при создании контекста приложения функцией IrbisAppInit.

Пример непосредственно из кода приложения Web-ИРБИС64+PHP (см callback.inc).

Сигнатура функции, которая будет вызываться из языка форматирования должна содержать следующие параметры

resource $sp – экземпляр открытой базы данных

int $Curr\_Shelf – номер «полки», то есть буфера, в котором содержится запись (всего их 7)

int $OCC\_Exit – номер текущего повторения поля для повторяющихся групп (циклов по повторениям полей)

string $params – строка параметров

bool &$handled – флаг с отметкой, что вызов обработан на стороне PHP.

Последний параметр используется только при перехвате функций unifor и umarci, для обычных функций его можно не указывать.

<?php

//функция, которую будет вызывать PHP (перехват unifor)

function php\_unifor($sp, $Curr\_Shelf, $OCC\_Exit, $params, &$handled)

{

//заменяем функция trim из ex64

if (strpos($params, '+8ex64,trim,') !== FALSE)

{

$handled = true;

return trim(substr($params, 12));

}

}

//еще одна функция, которую будем вызывать из PHP, вывод текущей даты в нужном формате

function php\_date($sp, $Curr\_Shelf, $OCC\_Exit, $params)

{

return date('l \t\h\e jS');

}

//список функций, которые будут вызываться на стороне PHP

$irbis\_callbacks = array

(

"UF" => "php\_unifor",

"UNIFOR" => "php\_unifor",

"PHP\_DATE" => "php\_date",

);

//инициализация контекста приложения

$ctx = IrbisAppInit('e:/xampp\_2/htdocs/web/irbis\_server\_ft\_php.ini', $irbis\_callbacks);

if ($ctx === FALSE)

{

trigger\_error("failed to init application context", E\_USER\_WARNING);

exit;

}

//включение интерпретатора xpft (не обязательно, по умолчанию выключен)

irbis\_enable\_xpft($ctx, TRUE);

//открытие базы данных

$sp = IrbisInit($ctx, "ibis");

If ($sp === FALSE)

{

trigger\_error("failed to init application context", E\_USER\_WARNING);

exit;

}

//загрузка записи с MFN 1 на полку 0 (всего их 7, нумерация с 0)

IrbisRecord($sp, 0, 1);

//компиляция формата с вызовом наших функций

//обратите внимание, что в функцию нужно обязательно передать параметр, даже если он пустой

//это специфика синтаксиса

Irbis\_InitPFT($sp, "&uf('+0'),&uf('+8ex64,trim,',v100),' current date: ', &php\_date('')");

//выполнение формата ($result – код ошибки или 0, текст в выходном параметре)

$fmt = '';

$result = Irbis\_Format($sp, 0, 1, $fmt);

echo $fmt;

//закрытие ресурсов

IrbisClose($sp);

IrbisAppClose($ctx);

?>

### Использование блоков PHP во фреймах интерфейсной части Web-ИРБИС64

Для блоков PHP предусмотрен синтаксис <?php ?>

Пример файла \*.frm

<html><head><title>ИРБИС64+ Электронная библиотека</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<link rel="stylesheet" href="<--HTTP\_PATH-->/css/styles.css">

<link rel="SHORTCUT ICON" href="<--HTTP\_PATH-->/images/irbis64.ico">

</head>

<body>

<?php

//вызов блока на языке PHP

echo "I am from php<br/>\n";

?>

<?

'I am from pft'

?>

</body>

</html>

### Вызов кода PHP во время вторичной обработки

В логике работы шлюза Web-ИРБИС64 заложена одна малоизвестная особенность.

После выполнения форматов приложение просматривает получившийся текст, чтобы обрабатывать шаблоны вроде <--HTTP\_PATH-->, <--DATABASE--> и так далее. Если формат сформирует текст, который будет содержать тег <?php ?>, то данный тег будет обработан интерпретатором PHP.

### Отладка ошибок в форматах

Интерпретатор xpft выводит ошибки в форматах в поток ошибок PHP, который может быть перенаправлен в отдельный файл или в лог файл сервера. При работе с PHP в режиме разработчика (например, при использовании php.ini-development в качестве php.ini) ошибки в форматах выводятся в тело документа. Кроме этого, ошибки времени разбора формата и ошибки времени выполнения пишутся в лог файл xpft\_error.log. В windows данный файл находится рядом с исполняемым файлом \*.exe, который использует PHP. В Linux данный файл находится в директории /tmp

### Расширенный синтаксис xpft

Интерпретатор XPFT предоставляет дополнительные возможности языку форматирования ИРБИС: дополнительные функции, расширенный доступ к повторениям полей и подполей, синтаксис прямого присваивания полей записи и глобальных переменных, числовые и строковые переменные, case оператор и оператор цикла while. Синтаксис XPFT основан на языках форматирования J-ISIS и CISIS и адаптирован для поддержки форматов ИРБИС.

**Числовые и строковые переменные**

Числовые переменные обозначаются e0…eN, в текущей реализации доступны переменные от e0 до e9. Так как синтаксис присваивания числовых переменных не содержит специальных разделителей (в отличие от синтаксиса присваивания строковых переменных), то после оператора присваивания следует ставить разделитель запятую.

Присваивание e0 := 1, e1 := (2 / 1),

Чтение f(e0 + e1, 0, 0)

Следующий пример иллюстрирует ошибку синтаксиса, когда отсутствует разделитель между командами.

e0 := 1 /, 'текст'

Здесь парсер посчитает 1 / частью одного выражения и выдаст ошибку.

Корректный пример.

e0 := 1, /, 'текст'

Строковые переменные обозначаются s0…sN, в текущей реализации доступны переменные от s0 до s9. Выражение, формирующее значение в операторе присваивания обязательно нужно заключать в скобки. Скобки являются ограничителями цепочки команд, формирующих строковое выражение.

Присваивание s1 := ('текст'),

Чтение s1,

**Доступ к повторениям полей и подполей**

Синтаксис доступа к повторениям полей аналогичен синтаксису классического языка форматирования ИРБИС, но несколько усовершенствован. В спецификации диапазона можно использовать также динамические выражения.

Несколько примеров.

v1[1] – вывод первого повторения поля 1.

v1[2..] – вывод повторений со второго до последнего повторения поля 1.

e0 := 1, v1[e0..2] – вывод повторений поля 1 с использованием динамического выражения.

v1[2]^a – вывод подполя A второго повторения поля 1 (данный синтаксис выводит только первое повторение).

v1[2]^a[2] – вывод второго повторения подполя A второго повторения поля 1.

v1[2]^a[2..] – вывод повторений подполя A со второго до последнего второго повторения поля 1

v1[LAST] – вывод последнего повторения поля 1

v1[LAST..LAST] – вывод последнего повторения поля 1

v1[2..LAST] – вывод повторений поля 1 со второго до последнего

v1[2]^a[LAST] - вывод последнего повторения подполя A второго повторения поля 1.

Ключевое слово LAST означает последнее повторение поля или подполя.

Примечания.

При указании в нижней границе диапазона значений меньше 1 указанные повторения игнорируются. В верхней границе диапазона значения меньше 1 трактуются как значение верхней границы.

Пример.

v1[2]^a[-1..] – данный формат выведет повторения с первого до последнего, так как повторения -1 .. 0 будут отброшены.

v1[2]^a[1..-1] – то же самое, что v1[2]^a[1..] или v1[2]^a[1..LAST]

**Работа с повторяющимися группами**

При указании диапазона повторений **поля** в повторяющейся группе, будет выводиться повторение равное **значению нижней границы диапазона**. Для доступа к текущему повторению повторяющейся группы используйте оператор iocc.

Вывод фиксированного повторения также вызывает еще одну итерацию в стиле ИРБИС и может приводить к бесконечным циклам.

Пример. Бесконечный цикл до отсечки

g1 := ('1',/,'2',/,'3'), ( g[2..LAST], /)

Вывод формата

2

2

2

…

Пример. Управляемый бесконечный цикл.

g1 := ('1',/,'2',/,'3'), ( if iocc <= 5 then g1[1] else 'good bye' fi, /)

Вывод формата

1

1

1

1

1

good bye

При выводе повторений подполей счетчик повторяющейся группы не используется. Для этого можно использовать обычный оператор цикла **while**.

Комплексный пример.

e0:=0, g1 := ('1^as^aj^ak',/,'2^a2s^a2j^a2k',/,'3^a3s^a3j^a3k'),(g1[iocc + e0..]^a[2..], e0 := e0 + 1,/)

Формат выполняет следующие действия

**e0:=0** – переменная e0 устанавливается в 0

**g1 := ('1^as^aj^ak',/,'2^a2s^a2j^a2k',/,'3^a3s^a3j^ak3')** – присваивание значения глобальной переменной G1 по результатам выполнения формата. Строка результата разбивается на список строк, каждая непустая строка в списке становится отдельным повторением. После выполнения данной команды содержимое G1 будет таким:

1 повторение: 1^as^aj^ak

2 повторение: 2^a2s^a2j^a2k

3 повторение: 3^a3s^a3j^ak3

И, собственно, сам цикл вывода значений **(g1[iocc + e0..]^a[2..], e0 := e0 + 1,/)**

Здесь последовательно выводятся повторения со второго до последнего подполя A глобальной переменной G1. При этом, номер выводимого повторения глобальной переменной вычисляется как сумма счетчика повторений повторяющейся группы (iocc) плюс значение переменной e0 (берется только нижняя граница диапазона). Значение переменной e0 модифицируется после вывода значения.

На первой итерации значение переменной e0 будет 0, значение счетчика повторяющейся группы 1, то есть будет выведено 1 повторение глобальной переменной G1 (0 + 1 = 1). На второй итерации повторяющейся группы значение переменной e0 будет 1, значение счетчика повторяющейся группы 2, то есть будет выведено третье повторение глобальной переменной G1 (1 + 2 = 3).

Вывод формата:

jk

3j3k

**Синтаксис присваивания значений полей, подполей записи и глобальных переменных**

Присваивание строковых выражений должно записываться так имя := (формат).

Скобки служат ограничителями группы команд.

Примеры

s0 := (''),s0 := (s0,'123'), s0 – установка сначала пустой строки в переменную s0, затем конкатенации старого значения s0 и строки 123 и затем вывод переменной s0

g100:=(''),g100:=(v25[1..4]^a[1]) – установка значения глобальной переменной с номером 100 путем прямого копирования значений поля.

В тех случаях, когда производится присваивание значений полей и глобальных переменных и новое значение является выражением поля (т.е. поле или глобальная переменная без дополнительных команд), то значения нужных повторений полей и подполей копируются напрямую без выполнения формата.

Тот же пример, но уже с выполнением формата и копированием результатов в глобальную переменную (каждая непустая строка становится отдельным повторением).

g100:=(''),g100:=((v25[iocc..4]^a[1] /))

v31:=(v25) – прямое копирование значений поля в другое поле

**Работа в повторяющихся группах**

При присваивании полей записи и глобальных переменных в повторяющейся группе со спецификацией повторения действует тот же принцип обработки повторений, что и при выводе.

Если задана спецификация повторения, повторение будет вычисляться по нижней границе диапазона. Для задания номера повторения в спецификации по счетчику повторяющейся группы следует использовать команду iocc.

v31:=('1 occ'),(v31[iocc + 1]:=(v25)) – копирование значений поля со «смещением» повторений. Здесь первым повторением поля 31 будет строка 1 occ, далее со второго повторения встанут копии повторений 25 поля.

Присваивание значений со спецификацией повторений

g100:=(''),g100^a[2]:=('12345'),g100 – здесь во второе повторение подполя a глобальной переменной записывается строка 12345, результат выполнения формата ^a^a12345

Заполнение подполей элементами записи

В данном случае при вставке некого текста в диапазон подполей подполя заполняются отдельными элементами. Элементом записи считается текст до первого подполя и отдельные значения подполей. Обратите внимание, что коды подполей в строке - источнике не имеют значения.

g100:=(''),g100^a[2..3]:=('12345^v6789'),g100 – выводит ^a^a12345^a6789

Тот же пример, но в данном случае элементов в источнике больше, чем требуется. В данном случае лишние элементы отбрасываются.

g100:=(''),g100^a[2..3]:=('12345^v6789^v00012'),(g100 /)

Формат выводит ^a^a12345^a6789

Тот же пример, но с несколькими повторениями поля в источнике. В полях назначения заполняется фиксированное количество повторений подполей.

g100:=(''),g100^a[2..3]:=('12345'/'6789'),(g100 /)

Данный формат выведет 2 повторения поля, в каждом из которых заполнено второе и третье повторения подполя

^a^a12345^a

^a^a6789^a

Аналогичный пример с заполнением подполей. Только теперь требуется заполнить более широкий диапазон в записи назначения.

Недостающие элементы заполняются пустыми строками.

g100:=(''),g100^a[2..4]:=('^v6789^v00012'),(g100 /)

Вывод ^a^a6789^a00012^a

Замена фрагментов текста внутри подполей

Наиболее сложный пример.

Если требуется заменить фрагмент строки в поле можно воспользоваться следующим синтаксисом. При этом если требуемое повторение поля или подполя не найдено в записи назначения или оно пустое, значение будет просто добавлено (нечего заменять).

Если значение присутствует, будет заменен фрагмент строки. При этом полностью пустые повторения полей не добавляются (правило ИРБИС).

g100:=(''),g100[3..4]^a[2..3]\*2.5:=('12345'/'6789'),(g100 /)

Данный формат пытается заполнить фрагмент подстроки, начиная с третьего символа (нумерация с нуля) длиной пять символов во втором и третьем повторениях подполя A в третьем и четвертом повторениях глобальной переменной 100. Так начальное количество повторений в глобальной переменной 0, то первое и второе повторения не добавляются. Так как в добавляемых двух повторениях нет подполя A, то нужные повторения подполя A инициализируются. При этом в исходных данных есть только текст без подполей, поэтому реальное значение появляется только во втором повторении подполя A.

Формат выводит

^a^a12345^a

^a^a6789^a

Аналогичный пример, но когда уже есть что заменить в записи назначения.

g100:=('^a^a000000000'),g100[1..2]^a[2..3]\*2.5:=('12345'/'6789'),(g100 /)

Здесь начальное значение глобальной переменной 100 содержит одно повторение, в котором есть 2 повторения поля A. Требуется заменить фрагмент строки с третьего символа длиной 5 символов во втором и третьем повторениях подполя A.

В имеющемся первом повторении поля во втором повторении подполя нужный фрагмент найден, второе повторение поля просто добавляется с инициализацией нужных повторений подполя A.

Формат выводит

^a^a001234500^a

^a^a6789^a

Установка полей записи в цикле

g100:=(''),e1:=0,while e1 < 5 ( e1:=e1+1, g100[e1]:=(f(e1,0,0)) ),(g100/)

Здесь выполняется цикл с итерационной переменной в диапазоне от 0 до 4, всего 5 итераций.

Строковое представление переменной записывается отдельным повторением в глобальную переменную 100.

Формат выводит

1

2

3

4

5

Дополнительные функции, которые не были представлены ранее в языке форматирования ИРБИС

|  |  |
| --- | --- |
| **@** | **Выполнение внешнего файла формата в стиле CISIS и J-ISIS** |

|  |  |
| --- | --- |
| *Синтаксис:* | **@**имя файла без расширения |
| *Описание:* | Компилирует и выполняет формат из указанного файла. |
| *Примечания:* | Поиск файла формата производится в соответствии с правилами ИРБИС. Сначала поиск в пользовательском депозите, затем в зависимости от настроек в INI файле в депозите и в каталоге базы данных. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **@**test,v20, | |
|  |  |
|  | |  |  | | --- | --- | | 2 | if v1='L' then **@**large, fi, | |

|  |  |
| --- | --- |
| **cat(file)** | **Вывести содержимое файла** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **cat(формат**) |
| *Описание:* | Выводит содержимое файла, имя которого формируется форматом. Функциональность несколько отличается от аналогичной функции в языке форматирования CISIS. Так, передаваемое имя файла ищется по умолчанию в каталоге базы данных. Для указания места, где искать файл в начале имени файла должен стоять номер пути в стиле ИРБИС. |
| *Примеры:* | |  |  | | --- | --- | | 1 | cat('ibis.par') вывод файла ibis.par из каталога DATAI системы ИРБИС | |
|  | |  |  | | --- | --- | | 2 | cat('11,d:/test.txt') вывод файла по абсолютному пути | |
|  | |  |  | | --- | --- | | 3 | cat(v101), | |

|  |  |
| --- | --- |
| **citype(string)** | **Тип строкового выражения** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **citype(формат)** |
| *Описание:* | Возвращает тип строкового выражения: A – если строка содержит только буквенные символы или пробелы N – если строка содержит только числовые символы (0-9) X – во всех остальных случаях |
| *Примечания:* | Формат должен формировать строковое выражение. Функция работает с поддержкой utf-8 |
| *Пример:* | |  |  | | --- | --- | | 1 | if **citype(**v1**)**='N' then f(val(v1),3,2)/ else v1/, fi, | |
|  | |  |  | | --- | --- | | 2 | if s(**citype(**v1**)**,**citype(**v2**)**,**citype(**v3**)**)<>'AAA' then 'Invalid character type detected'/, fi, | |
| *См также* | **type** |

|  |  |
| --- | --- |
| **date** | **Вывод текущей даты** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **date** **date**(ключевое слово) |
| *Описание:* | Выводит текущую системную дату. При использовании без параметров выводит дату в формате: **yyyymmdd hhmmss w nnn** *где:*    yyyy = год     mm = месяц     dd = день     hh = час    mm = минута    ss = секунда     w = день недели (0-6)     nnn = день от начала года. |
| *Параметры:* | Ключевое слово, передаваемое в параметра может быть **DATETIME** или **DATEONLY**, также доступны форматы 1,2 и 3 |
| *Примечание:* | **DATETIME** отображает дату в европейском формате плюс текущее время (dd/mm/yy hh:mm:ss), а **DATEONLY** отображает только дату.  Если передан параметр 1 дата выводится в формате mm-dd-yy hh:mm:ss. Если передан параметр 2, то mm-dd-yy. Если передан параметр 3, то hh:mm:ss. |
| *Примеры:* | |  |  | | --- | --- | | 1 | 'Today is ',**date**, | |
|  | |  |  | | --- | --- | | 2 | 'Current date: ',**date(DATEONLY)**/, ,'Current time: ',mid(**date(DATETIME)**,10,8)/, | |

|  |  |
| --- | --- |
| **datex(seconds)** | **Преобразование метки времени unix в дату** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **datex(формат)** |
| *Описание:* | Формат в параметрах должен формировать числовое значение, иначе будет ошибка синтаксиса. Преобразует значение, заданное форматом в дату в формате: **yyyymmdd hhmmss w nnn** *где:*    yyyy = год     mm = месяц     dd = день     hh = час    mm = минута    ss = секунда     w = день недели (0-6)     nnn = день от начала года. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **datex**(**seconds**('19800101')) | |
| *См также* | **seconds** |
|  |  |

|  |  |
| --- | --- |
| **db** | **Имя текущей базы** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **db** |
| *Описание:* | Выводит имя открытой базы. |
| *Примеры:* | |  |  | | --- | --- | | 1 | 'current db ',**db** | |
|  |  |

|  |  |
| --- | --- |
| **fmt(format)** | **Форматирование текущей записи в независимом контексте** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **fmt(**формат**)** |
| *Описание:* | Выполняет формат с текущей записью в независимом контексте. Это позволяет выполнять, например, форматы с повторяющимися группами внутри повторяющихся групп без использования **unifor 6** или **ref**. Строковые, числовые и глобальные переменные при этом доступны. |
| *Примеры:* | |  |  | | --- | --- | | 1 | ( v910**, fmt(** ( v200^a /) ), | |
|  |  |
|  |  |
| *См также:* | **s , ref** |

|  |  |
| --- | --- |
| **getenv(expression)** | **Вывод переменной окружения** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **getenv(**формат**)** |
| *Описание:* | Возвращает переменную окружения в виде строки. |
| *Примечания:* | Если формат не формирует корректное имя переменной окружения, значение не возвращается. |
| *Примеры:* | |  |  | | --- | --- | | 1 | 'Current path: ',**getenv(**'PATH'**)**, | |
|  | |  |  | | --- | --- | | 2 | (v1|=|,**getenv(**v1**)**/), | |
| *См также:* | **putenv** |

|  |  |
| --- | --- |
| **instr(string1,string2)** | **Поиск подстроки** |

|  |  |
| --- | --- |
| *Тип функции:* | числовая |
| *Синтаксис:* | **instr(**формат-1,формат-2**)** – синтаксис CISIS  **instr(**формат-1;формат-2**)** – синтаксис J-ISIS |
| *Описание:* | Возвращает позицию первого вхождения строки, которую формирует формат-2 в строке, которую формирует формат-1. Если строка не найдена, функция вернет 0. |
| *Примечания:* | Оба формата формат-1 и формат-2 должны генерировать строковые значения, в противном случае будет ошибка синтаксиса. В тех случаях, когда в параметрах используются форматы, состоящие из нескольких элементов, может помочь функция ***s***. Вариант синтаксиса J-ISIS отличается тем, что в форматах можно использовать цепочку команд без использования функции **s** как контейнера. |
| *Примеры:* | |  |  | | --- | --- | | 1 | if **instr(**v5,'ab'**)**>0 then v5/, fi, | |
|  | |  |  | | --- | --- | | 2 | if **instr(***s(|'|v1|'|)***,***v5***)**>0 then v1, fi, | |
|  | |  |  | | --- | --- | | 3 | left(v18,**instr(**v18,'.'**)**-1), | | 4 | f(instr('1', '2', '3'; '1', '2'),0,0) | |

|  |  |
| --- | --- |
| **iocc** | **Номер текущего повторения** |

|  |  |
| --- | --- |
| *Тип функции:* | Числовая |
| *Синтаксис:* | **iocc** |
| *Описание:* | Возвращает номер текущего повторения в повторяющейся группе (начиная с 1), если текущий формат выполняется вне повторяющейся группы вернет 0. |
| *Примеры:* | |  |  | | --- | --- | | 1 | ("Author: "v1/, ,if **iocc** > 3 then 'et all',break, fi), | |
|  | |  |  | | --- | --- | | 2 | (f(**iocc**,3,0),|.|v10/), | |
| *См также:* | **nocc** |

|  |  |
| --- | --- |
| **l(key) l([inverted file],key)** | **Поиск ключа в словаре** |

|  |  |
| --- | --- |
| *Тип функции:* | Числовая |
| *Синтаксис:* | **l(**термин**)** **l(**[имя бд]термин**)**  **l->ИМЯ БД(**термин**)** |
| *Описание:* | Возвращает **MFN** первой ссылки (если есть), в качестве поискового термина используется результат выполнения формата в параметрах. Поиск производится по текущей базе данных. Дополнительно имя базы данных можно указать статично с использованием синтаксиса ->ИМЯ, либо с помощью формата [‘формат’] (имя может быть динамическим). Указание имени базы данных в классическом языке форматирования ИРБИС отсутствует |
| *Примечания:* | Если термин не найден в базе, функция вернет 0. Если имя базы данных задано форматом, он должен возвращать строку, иначе будет ошибка синтаксиса. Эта функция часто используется вместе с **ref**. |
| *Примеры:* | |  |  | | --- | --- | | 1 | if **l(**v15**)**<> 0 then |Term: |v15, fi, | |
|  | |  |  | | --- | --- | | 2 | ref(**l(['books']**v1,'-',v2**)**,v10/), | | 3 | ref->RDR(**l->RDR(**v1,'-',v2**)**,v10/), | |
| *См также:* | **ref** |

|  |  |
| --- | --- |
| **lr((key),from,to) lr([inverted file],(key),from,to)** | **Поиск ключа в словаре** |

|  |  |
| --- | --- |
| *Тип функции:* | Числовая |
| *Синтаксис:* | **lr**((термин))  **lr**((термин),номер первой ссылки,номер последней ссылки) **lr**([имя бд](термин))  **lr**([имя бд](термин), номер первой ссылки,номер последней ссылки)  **lr**->ИМЯ БД((термин))  **lr**->ИМЯ БД((термин), номер первой ссылки,номер последней ссылки) |
| *Описание:* | Возвращает список на термин (список MFN), в качестве поискового термина используется результат выполнения формата в параметрах. Поиск производится по текущей базе данных. Дополнительно имя базы данных можно указать статично с использованием синтаксиса ->ИМЯ, либо с помощью формата [‘формат’] (имя может быть динамическим). Можно указать часть диапазона в необязательных параметрах после термина. Формат, формирующий термин должен быть заключен в скобки. |
| *Примечания:* | Если имя базы данных задано форматом, он должен возвращать строку, иначе будет ошибка синтаксиса. Функцию можно использовать вместе с **npst** и **ref**. |
| *Примеры:* | |  |  | | --- | --- | | 1 | ref(**lr**((v10)),v1,v2) | |
|  | |  |  | | --- | --- | | 2 | ref(**lr**((v10),3,7),v1,v2) | | 3 | ref->RDR(**lr**->RDR((v1,'-',v2)),v10/), | |
| *См также:* | **ref,npst,l** |

|  |  |
| --- | --- |
| **left(string,length)** | **Подстрока слева** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **left(**формат-1,формат-2**)** – синтаксис CISIS  **left(**формат-1;формат-2**)** – синтаксис J-ISIS |
| *Описание:* | Возвращает строку, которая содержит левую часть строки, формируемую форматом формат-1. Формат формат-2 означает количество символов слева, которые вернет функция. |
| *Примечания:* | Формат-2 должен возвращать число, иначе будет ошибка синтаксиса. Если количество символов, возвращаемое форматом формат-2 больше, чем длина строки, функция вернет строку целиком. Если количество символов меньше, либо равно нулю, вернется пустая строка. Функция интерпретирует исходную строку как utf-8. Вариант синтаксиса J-ISIS отличается тем, что в первом формате можно использовать цепочку команд без использования функции **s** как контейнера. |
| *Примеры:* | |  |  | | --- | --- | | 1 | if **left(**v1^n,2**)**='Ma' then v1^n/, fi, | |
|  | |  |  | | --- | --- | | 2 | **left(**v1,instr(v1,'.')-1**)**, | | 3 | **left(**'1', '2', '3'; 2) | |
| *См также:* | **right** **mid** |

|  |  |
| --- | --- |
| **ltrim(string[,string])** | **Удаление пробельных символов слева** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **ltrim(формат1)**  **ltrim(формат1,формат2)** |
| *Описание:* | Удаляет пробельные символы, либо символы, заданные форматом формат2 слева в строке, формируемой формат1. Форматы формат1 и формат2 могут быть составлены из нескольких выражений.  Поддерживается вариант, когда разделителем параметров служит запятая и точка с запятой. |
| *Примечания:* | В том случае, если в параметрах представлены ровно 2 выражения, разделенных запятой, функция считает, что первое — это формат1, второе — формат2. Если выражений, разделенных запятой одно или больше 2 или, то все они трактуются, как формат1. В случае использования составных выражений лучше использовать точку с запятой в качестве разделителя. |
| *Пример:* | |  |  | | --- | --- | | 1 | **ltrim**('я формат1', 'а я формат2') | |
|  | |  |  | | --- | --- | | 2 | **ltrim**('все мы',' вместе ','формат1') | | 3 | **ltrim**('вот так',' правильно ','использовать составные выражения' ; ' ') | |
| *См также* | **trim,rtrim** |

|  |  |
| --- | --- |
| **maxmfn** | **Максимальный mfn + 1** |

|  |  |
| --- | --- |
| *Тип функции:* | числовая |
| *Синтаксис:* | **maxmfn** |
| *Описание:* | Возвращает число, соответствующее максимальному MFN в текущей базе данных + 1. Может использоваться как в качестве команды вывода (аналогично команде mfn), так в качестве числового выражения. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **maxmfn** | |
|  | |  |  | | --- | --- | | 2 | if **maxmfn >** 1 then'have records' fi | |  |  | |
| *См также:* | **mfn** |

|  |  |
| --- | --- |
| **mid(string, start, length)** | **Подстрока** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **mid(**формат-1,формат-2,формат-3**)** – синтаксис CISIS  **mid(**формат-1;формат-2;формат-3**)** – синтаксис J-ISIS |
| *Описание:* | Возвращает строку, содержащую указанное количество символов с указанной позиции. Формат-1 формирует исходную строку, формат-2 начальную позицию подстроки (начиная с 1), формат-3 количество символов. Форматы 2 и 3 должны возвращать числовое значение, иначе будет ошибка синтаксиса. |
| *Примечания:* | Позиция начала подстроки больше, чем длина исходной строки, функция вернет пустую строку. Если позиция подстроки меньше или равна 0, функция вернет подстроку с первого символа. Функция интерпретирует исходную строку как utf-8. Вариант синтаксиса J-ISIS отличается тем, что в первом формате можно использовать цепочку команд без использования функции **s** как контейнера. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **mid(**v2,2,80**)**, | |
|  | |  |  | | --- | --- | | 2 | **mid(**v1,instr(v1,'key'),size(v1)**)/**, | | 3 | **mid(**'1', '2', '3'; 1; 2) | |
| *См также:* | **right** **left**  **ss** |

|  |  |
| --- | --- |
| **newline(string)** | **Замена экранированных спец символов** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **newline(формат)** |
| *Описание:* | Заменяет экранированные символы \r, \n, \t, \b, \n, \0 на реальные символы |
| *Примечания:* | Спец символы с ASCII кодами ниже 32 невозможно использовать в литералах напрямую, парсер их просто игнорирует. Для этого можно использовать экранированные строки с помощью функции newline. Символ \0 означает конец строки. Дальше этого символа строка обрезается. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **newline**(if v151='unix' then '\n' else '\r\n' fi), | |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **nocc(field)** | **Количество повторений поля** |

|  |  |
| --- | --- |
| *Тип функции:* | Числовая |
| *Синтаксис:* | **nocc(**поле либо глобальная переменная**)** |
| *Описание:* | Возвращает количество повторений поля, либо глобальной переменной, определяемой параметром. |
| *Примечание:* | В спецификации поля можно использовать код подполя, номер повторения поля или подполя, спецификацию подстроки |
| *Примеры:* | |  |  | | --- | --- | | 1 | if **nocc(**v3**)**> 10 then 'Too many occurrences.'/, fi, | |
|  | |  |  | | --- | --- | | 2 | 'There are ',f(**nocc(**v20**)**,2,0),' authors.'/, | | 3 | g920 := ('1^a123'/'2'), f(nocc(g920^a\*1),1,1) | | 4 | g920 := ('1^a123^a321'/'2'), f(nocc(g920^a[2]\*1),1,1) | |
| *См также:* | **iocc** |

|  |  |
| --- | --- |
| **npost(key) npost([inverted file],key)** | **Количество ссылок на термин** |

|  |  |
| --- | --- |
| *Тип функции:* | числовая |
| *Синтаксис:* | **npost(**термин**)** **npost(**[имя бд],термин**)**  **npost->ИМЯ БД(**термин**)**  синоним данной функции npst  **npst(**термин**)** **npst(**[имя бд],термин**)**  **npst->ИМЯ БД(**термин**)** |
| *Описание:* | Возвращает количество ссылок на термин. Термин для поиска формируется форматом, передаваемым в параметре. Поиск производится в текущей базе данных. Можно указывать другую базу данных с помощью синтаксиса ->ИМЯ БАЗЫ или [ИМЯ БАЗЫ]. Во втором случае допускается использование динамических выражений. |
| *Примеры:* | |  |  | | --- | --- | | 1 | if **npost(**v1**)**> 1 then 'duplicate key ',v1,' found'/, fi, | |
|  | |  |  | | --- | --- | | 2 | 'There are ',f(**npost([**'RDR'**]**v20**)**,3,0),'keys for ',v20,'. '/, | |
| *См также:* | **l** |

|  |  |
| --- | --- |
| **proc(field update format)** | **Обновление полей в стиле CISIS** |

|  |  |
| --- | --- |
| *Тип функции:* | Без возвращаемого значения |
| *Синтаксис:* | **proc(формат)** |
| *Описание:* | Добавляет или удаляет поля в текущей записи. Формат в параметре формирует строку с данными. |
| *Примечания:* | Спецификация строки с данными. Строка должна содержать набор команд **d** (удалить) или **a**(добавить) and **h** (добавить). Команды **d** (удалить) должны быть перед любыми другими командами. Доступные команды: **d\***   -   удаляет все поля **d**метка поля   -   удаляет все повторения указанного поля **d**метка поля**/**повторение   -  удаляет указанное повторение поля **a**метка поля**#строка#**   -   добавляет новое повторение поля **h**метка поля **n** строка   -   добавляет n символов строки как новое повторение поля Разделитель **#** может быть любым нецифровым символом. В команде **h** в качестве разделителя метки поля, количества символов и строки можно использовать любой также любой символ. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **proc(**'d70',|a10#|v70|#|**)**, | |
|  | |  |  | | --- | --- | | 2 | **proc(**if v24\*0.4 = 'Tech' then 'd\*', fi**)**, | |

|  |  |
| --- | --- |
| **putenv(expression)** | **Установка переменной окружения** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **putenv(**формат**)** |
| *Описание:* | Устанавливает значение переменной окружения. Возвращает устанавливаемое значение |
| *Примечания:* | Установленное значение действительно в рамках текущего процесса |
| *Примеры:* | |  |  | | --- | --- | | 1 | **putenv(**'TEST=test'**)**,getenv('TEST'), | |
|  |  |
| *См также:* | **getenv** |

|  |  |
| --- | --- |
| **ref(mfn, format) ref([master file]mfn, format)** | **Форматирование записи по MFN** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **ref(**MFN, формат**)** **ref->ИМЯ БД(**MFN, формат**)**  **ref(**['ИМЯ БД'] MFN, формат**)** |
| *Описание:* | Выполняет формат с записью, которая загружается по MFN. Выражение, которое формирует MFN может быть динамическим. Запись загружается из текущей базы, возможно указать другую базу данных. |
| *Примечания:* | Для поиска MFN можно использовать функцию L. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **ref(**l(v3),v1/,v2/,v3/**)**, | |
|  | |  |  | | --- | --- | | 2 | if **ref(**['account']l(['user']v2),v4**)**='active' then |Name: |v10/, fi, | |
|  | |  |  | | --- | --- | | 3 | (if p(v99) then **ref(**[v99]1,v30/**)**, fi), | |
| *См также:* | **l, fmt** |

|  |  |
| --- | --- |
| **replace(string1, string2, string3)** | **Замена подстроки** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **replace(**формат-1, формат-2, формат-3**)** – синтаксис CISIS  **replace(**формат-1; формат-2; формат-3**)** – синтаксис J-ISIS |
| *Описание:* | Возвращает строку, формируемую форматом формат-1, заменяя подстроку, формируемую форматом формат-2 на подстроку, формируемую форматом формат-3. |
| *Примечания:* | Если строка, формируемая форматом формат-2 пустая, или подстрока не найдена, вернется строка, формируемая форматом формат-1, без изменений. При поиске учитывается регистр символов. Вариант синтаксиса J-ISIS отличается тем, что в форматах можно использовать цепочки команд без использования функции **s** как контейнера. Это связано с тем, что разделитель команд точка с запятой сам по себе уникальный и в языке форматирования не встречается. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **replace(**'Mary And John','And','and'**)/**, | |
|  | |  |  | | --- | --- | | 2 | if **replace(**v1^a,'01x','01X'**)**= '894501X' then v1^n/, fi, | |
|  | |  |  | | --- | --- | | 3 | **replace**(s(v304,v333),',',', ')/, | |
|  | |  |  | | --- | --- | | 4 | **replace(**s(if v415='spanish' then v299 else 'none' fi),v1,v759**)/**, | | 5 | **replace(**'123','3'; '1', '2'; '-', -') | |

|  |  |
| --- | --- |
| **right(string, length)** | **Подстрока справа** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **right(**формат-1,формат-2**)** – синтаксис CISIS  **right(**формат-1;формат-2**)** – синтаксис J-ISIS |
| *Описание:* | Возвращает правую часть строки, формируемой форматов формат-1. Формат-2 определяет количество символов. Формат-2 должен возвращать число, иначе будет ошибка синтаксиса. |
| *Примечания:* | Если количество символов больше, чем длина строки, функция вернет строку целиком. Если количество символов меньше, либо равно 0, функция возвращает пустую строку. Функция интерпретирует исходную строку как utf-8. Вариант синтаксиса J-ISIS отличается тем, что в первом формате можно использовать цепочку команд без использования функции **s** как контейнера. |
| *Примеры:* | |  |  | | --- | --- | | 1 | if **right(**v1^n,1**)** = 'r' then v1^n/, fi, | |
|  | |  |  | | --- | --- | | 2 | **right**(v65,4)/, | | 3 | **right**('1', '2', '3'; 2) | |

|  |  |
| --- | --- |
| **rtrim(string[,string])** | **Удаление пробельных символов справа** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **rtrim(формат1)**  **rtrim(формат1,формат2)** |
| *Описание:* | Удаляет пробельные символы, либо символы, заданные форматом формат2 справа в строке, формируемой формат1. Форматы формат1 и формат2 могут быть составлены из нескольких выражений.  Поддерживается вариант, когда разделителем параметров служит запятая и точка с запятой. |
| *Примечания:* | В том случае, если в параметрах представлены ровно 2 выражения, разделенных запятой, функция считает, что первое — это формат1, второе — формат2. Если выражений, разделенных запятой одно или больше 2 или, то все они трактуются, как формат1. В случае использования составных выражений лучше использовать точку с запятой в качестве разделителя. |
| *Пример:* | |  |  | | --- | --- | | 1 | **rtrim**('я формат1', 'а я формат2') | |
|  | |  |  | | --- | --- | | 2 | **rtrim**('все мы',' вместе ','формат1') | | 3 | **rtrim**('вот так',' правильно ','использовать составные выражения' ; ' ') | |
| *См также* | **ltrim,trim** |

|  |  |
| --- | --- |
| **seconds(string)** | **Конвертация даты в метку времени unix** |

|  |  |
| --- | --- |
| *Тип функции:* | числовая |
| *Синтаксис:* | **seconds(**формат**)** |
| *Описание:* | Разбирает строку, формируемую форматом как дату в формате yyyymmdd [hhmmss] и преобразует ее в метку времени unix. Аналогична функции mktime в PHP. |
| *Примеры:* | |  |  | | --- | --- | | 1 | f(seconds('19801107'),0,0) | |
|  | |  |  | | --- | --- | | 2 | f(seconds('19871107 235959'),0,0) | |
| См также | **datex** |

|  |  |
| --- | --- |
| **select … case … elsecase … endsel** | **Условный оператор** |

|  |  |
| --- | --- |
| *Синтаксис:* | **select** формат-условие **case** опция-1: формат-1 **case** опция-2: формат-2 **case** опция-n: формат-n [**elsecase** формат-0] **endsel** |
| *Описание:* | Выполняет формат-условие и сравнивает результат с каждой опцией **case** (опция-1, опция-2…опция-n). Если опция совпадает с результатом, соответствующий ей формат будет выполнен (формат-1, формат-2…формат-n), в противном случае выполняется формат формат-0 в **elsecase** (если определен). |
| *Примечания:* | Формат-условие должен генерировать строковое либо числовое значение. Если формат-условие создает строковое значение, все опции должны также возвращать строковые значения. Если формат-условие создает числовое значение, все опции должны также быть числовыми значениями. Опции могут формироваться как статическими константами, так и динамическими выражениями. Если нужно объединить несколько команд в формате формат-условие используйте функцию s или fmt. |
| Примеры*:* | |  |  | | --- | --- | | 1 | **select** s(v5)  **case** f(0 + 1,1,1)**:** ,f(val(v5)/2,2,2)/,  **case** '2'**:** ,v5/,  **case** '3'**:** ,v6,'-',v1/,  **elsecase** ,|Error in field v5 = |v5/,  **endsel**, | |
|  | |  |  | | --- | --- | | 2 | **select** nocc(v7)  **case** 0: 'absent'/,  **case** 0 + 1: 'one occurrence'/,  **case** 2: 'two occurrences'/,  **elsecase** 'more than 2 occurrences'/,  **endsel**, | |
| *См также:* | **if … then … else … fi** |

|  |  |
| --- | --- |
| **size(string)** | **Длина строки** |

|  |  |
| --- | --- |
| *Тип функции:* | числовая |
| *Синтаксис:* | **size(**формат**)** |
| *Описание:* | Возвращает длину строки. |
| *Описание:* | Формат должен формировать строку, иначе будет ошибка синтаксиса |
| *Примеры:* | |  |  | | --- | --- | | 1 | if **size(**v10**)**> 76 then ' too heavy ', fi, | |
|  | |  |  | | --- | --- | | 2 | f(**size(**v10,v20),1,0**)**, | |

|  |  |
| --- | --- |
| **ss(start, length, string)** | **Подстрока** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **ss(**формат-1,формат-2,формат-3**)** |
| *Описание:* | Функция очень похоже на функцию **mid.**  Возвращает строку, содержащую указанное количество символов с указанной позиции. Формат-3 формирует исходную строку, формат-1 начальную позицию подстроки (начиная с 1), формат-2 количество символов. Форматы 1 и 2 должны возвращать числовое значение, иначе будет ошибка синтаксиса. Разница с mid состоит в том, что цепочки команд в формате формат-3 можно не объединять с помощью функции **s,** а использовать как есть. |
| *Примечания:* | Позиция начала подстроки больше, чем длина исходной строки, функция вернет пустую строку. Если позиция подстроки меньше или равна 0, функция вернет подстроку с первого символа. Функция интерпретирует исходную строку как utf-8. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **ss(**2,80, v2,v3**)**, | |
|  | |  |  | | --- | --- | | 2 | **ss(**instr(v1,'key'),size(v1), v1**)/**, | |
| *См также:* | **mid** |

|  |  |
| --- | --- |
| **system(expression)** | **Выполнение системной команды** |

|  |  |
| --- | --- |
| *Тип функции:* | не возвращает значения |
| *Синтаксис:* | **system(формат)** |
| *Описане:* | Выполняет системную команду, формируемую форматом. |
| *Примечания:* | Формат должен формировать строку, иначе будет ошибка синтаксиса. Вывод выполняемой команды не перехватывается. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **system(**'dir'**)**, | |
|  | |  |  | | --- | --- | | 2 | if p(v2) then **system(**'type ',v2**)**, fi, | |

|  |  |
| --- | --- |
| **tag** | **Номер метки поля** |

|  |  |
| --- | --- |
| *Синтаксис:* | **tag** |
| *Описание:* | Возвращает номер метки поля при выводе содержимого записи по команде v0 в повторяющейся группе |
| *Примечания:* |  |
| *Примеры:* | |  |  | | --- | --- | | 1 | (if p(v0) then f(**tag**,0,0),| |,v0/,fi) | |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **<text>строка</text>** | **Альтернативный синтаксис безусловного литерала** |

|  |  |
| --- | --- |
| *Синтаксис:* | **<text>**текст**</text>** |
| *Описание:* | Оператор ведет себя так же, как безусловный литерал |
| *Примечания:* | Можно применять вместо безусловного литерала в тех случаях, когда в тексте нужны символы одинарных кавычек. Символы с ASCII кодами ниже 32 также обрезаются, поэтому если нужно использовать в тексте литерала переводы каретки, символы табуляции и т.д. воспользуйтесь функцией **newline** |
| *Примеры:* | |  |  | | --- | --- | | 1 | **<text>**безусловный литерал с 'кавычками'**</text>** | |
|  |  |
|  | |  |  | | --- | --- | | 2 | **newline(<text>**безусловный литерал \n с 'кавычками'**</text>)** | |

|  |  |
| --- | --- |
| **trim(string[,string])** | **Удаление пробельных символов справа и слева** |

|  |  |
| --- | --- |
| *Тип функции:* | строковая |
| *Синтаксис:* | **trim(формат1)**  **trim(формат1,формат2)** |
| *Описание:* | Удаляет пробельные символы, либо символы, заданные форматом формат2 справа и слева в строке, формируемой формат1. Форматы формат1 и формат2 могут быть составлены из нескольких выражений.  Поддерживается вариант, когда разделителем параметров служит запятая и точка с запятой. |
| *Примечания:* | В том случае, если в параметрах представлены ровно 2 выражения, разделенных запятой, функция считает, что первое — это формат1, второе — формат2. Если выражений, разделенных запятой одно или больше 2 или, то все они трактуются, как формат1. В случае использования составных выражений лучше использовать точку с запятой в качестве разделителя. |
| *Пример:* | |  |  | | --- | --- | | 1 | **trim**('я формат1', 'а я формат2') | |
|  | |  |  | | --- | --- | | 2 | **trim**('все мы',' вместе ','формат1') | | 3 | **trim**('вот так',' правильно ','использовать составные выражения' ; ' ') | |
| *См также* | **ltrim,rtrim** |

|  |  |
| --- | --- |
| **type(string)** | **Тип строкового выражения** |

|  |  |
| --- | --- |
| *Тип функции:* | числовая |
| *Синтаксис:* | **type(число, формат)**  **type(шаблон, формат)** |
| *Описание:* | Проверяет тип строкового выражения, формируемого форматом на соответствие шаблону, в случае успеха, возвращает 1, иначе 0.  Если шаблон задан числом, то строка, формируемая форматом, проверяется на следующие типы:  1 – буквы и цифры  2 – только буквы  3 – только цифры  4 – целое число, в том числе со знаками + и -  5 – дробное число, в том числе в экспоненциальной форме, разделитель целой и дробной части точка  Если шаблон задан строкой, проверка ведется по следующим правилам:  A – буква  X – любой символ  9 – цифра  Все остальные символы сравниваются как есть. Шаблон должен быть той же длины, что и строка, формируемая форматом. |
| *Примечания:* | Формат должен формировать строковое выражение, шаблон может быть либо числовым выражением, либо строкой. Если строка, формируемая форматом пустая, функция вернет 1. Функция работает с поддержкой utf-8 |
| *Пример:* | |  |  | | --- | --- | | 1 | if **type(**1, v1**)** = 1 then f(val(v1),3,2)/ else v1/, fi, | |
|  | |  |  | | --- | --- | | 2 | if **type(**'AAAAXAAAAAAXAAAAAAA', 'ядра чистый изумруд'**)** <> 1 then 'это не Пушкин'/, fi, | |
| *См также* | **citype** |

|  |  |
| --- | --- |
| **ver** | **Версия записи** |

|  |  |
| --- | --- |
| *Тип функции:* | числовая |
| *Синтаксис:* | **ver** |
| *Описание:* | Возвращает версию текущей записи. Может использоваться как в качестве команды вывода (аналогично команде mfn), так в качестве числового выражения. |
| *Примеры:* | |  |  | | --- | --- | | 1 | **ver** | |
|  | |  |  | | --- | --- | | 2 | if **ver >** 1 then'record was edited' fi | |  |  | |
| *См также:* | **mfn** |

|  |  |
| --- | --- |
| **while** | **Оператор цикла** |

|  |  |
| --- | --- |
| *Синтаксис:* | **while** формат-условие ( формат ) |
| *Описание:* | Выполняет формат в скобках, пока формат-условие возвращает истину. |
| *Примечания:* | Формат-условие должен представлять булевское выражение, иначе будет ошибка синтаксиса. |
| Примеры*:* | |  |  | | --- | --- | | 1 | e1 := 1, **while** e1 <= 5 (f(e1,1,1), e1 := e1 + 1, /) | |
|  |  |
| *См также:* | **if … then … else … fi** |

### Функции модуля php\_irbis

|  |  |
| --- | --- |
| **resource IrbisAppInit(string inipath, [array callbacks])** | **Инициализация контекста приложения** |

|  |  |
| --- | --- |
| *Описание:* | Создание контекста приложения ИРБИС, который затем используется при открытии БД  Содержит кэш и разные статические данные  Один контекст может обслуживать несколько БД |
| *Параметры:* | inipath - Путь к ini файлу ИРБИС  callbacks – ассоциативный массив с описанием функций PHP, которые могут вызываться из форматов. Ключ – имя функции в языке форматирования, значение – имя функции в PHP |
| *Возвращаемое значение* | В случае успеха возвращается идентификатор ресурса, при ошибке FALSE |
|  |  |
| *См также:* | **IrbisAppClose** |

|  |  |
| --- | --- |
| **void IrbisAppClose (resource ctx)** | **Закрытие контекста приложения** |

|  |  |
| --- | --- |
| *Описание:* | Высвобождение контекста ИРБИС, созданного функцией IrbisAppInit |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | **IrbisAppInit** |

|  |  |
| --- | --- |
| **void irbis\_init\_DepositPath(resource ctx, string path)** | **Указание пути к депозиту** |

|  |  |
| --- | --- |
| *Описание:* | Указание пути к депозиту для контекста, созданного функцией IrbisAppInit,  если отличается от указанного в ini файле |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  path – путь к депозиту |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | **IrbisAppInit** |

|  |  |
| --- | --- |
| **void irbis\_init\_DataPath(resource ctx, string path)** | **Указание пути к базам данных** |

|  |  |
| --- | --- |
| *Описание:* | Указание пути к базам данных для контекста, созданного функцией IrbisAppInit, если отличается от указанного в ini файле |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  path – путь к базам данных |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | **IrbisAppInit** |

|  |  |
| --- | --- |
| **void irbis\_init\_ExecDir(resource ctx, string path)** | **Указание пути к каталогу ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Указание пути к каталогу ИРБИС для контекста, созданного функцией IrbisAppInit, если отличается от указанного в ini файле |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  path – путь каталогу ИРБИС |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | **IrbisAppInit** |

|  |  |
| --- | --- |
| **resource IrbisInit(resource ctx, string name)** | **Открытие базы данных** |

|  |  |
| --- | --- |
| *Описание:* | Открытие базы данных ИРБИС |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  name - Имя базы |
| *Возвращаемое значение* | Идентификатор ресурса, а в случае ошибки PHP FALSE |
|  |  |
| *См также:* | **IrbisAppInit, IrbisClose** |

|  |  |
| --- | --- |
| **void IrbisClose(resource sp)** | **Закрытие базы данных** |

|  |  |
| --- | --- |
| *Описание:* | Закрытие базы данных ИРБИС |
| *Параметры:* | sp - База ирбис, открытая функцией IrbisInit |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | **IrbisAppInit, IrbisInit** |

|  |  |
| --- | --- |
| **resource irbis\_ft\_space(resource sp)** | **Ссылка на ПБД** |

|  |  |
| --- | --- |
| *Описание:* | Возврат ссылки на вложенную полнотекстовую БД для каталога версии 2018 +. Если БД не содержит вложенной ПБД возвращает FALSE.  Данная ссылка должна быть также закрыта функцией IrbisClose |
| *Параметры:* | sp - База ирбис, открытая функцией IrbisInit |
| *Возвращаемое значение* | Идентификатор ресурса или FALSE в случае ошибки |
|  |  |
| *См также:* | **IrbisInit, IrbisClose** |

|  |  |
| --- | --- |
| **bool irbis\_is2018plus(resource sp)** | **Проверка формата БД** |

|  |  |
| --- | --- |
| *Описание:* | Проверка формата базы данных |
| *Параметры:* | sp - База ирбис, открытая функцией IrbisInit |
| *Возвращаемое значение* | Если база версии 2018+ возвращает TRUE, иначе FALSE |
|  |  |
| *См также:* | **IrbisInit** |

|  |  |
| --- | --- |
| **void irbis\_enable\_xpft(resource ctx, bool enabled)** | **Включение интерпретатора XPFT** |

|  |  |
| --- | --- |
| *Описание:* | Включение / выключение форматера XPFT для функции Irbis\_Format. По умолчанию форматер XPFT выключен. |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  enabled - включен или выключен форматер XPFT |
| *Возвращаемое значение* | Функция не возвращает значение |
|  |  |
| *См также:* | **Irbis\_InitPft, Irbis\_Format** |

|  |  |
| --- | --- |
| **void irbis\_use\_search\_cache(resource ctx, bool enabled)** | **Включение кэша поиска** |

|  |  |
| --- | --- |
| *Описание:* | Включение / выключение кэша поиска  для поисковых функций и unifor  По умолчанию кэш выключен |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  enabled - включен или выключен кэш поиска |
| *Возвращаемое значение* | Функция не возвращает значение |
|  |  |

|  |  |
| --- | --- |
| **int IrbisRecord(resource sp, int shelf, int mfn)** | **Загрузка записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Загрузка записи ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  mfn - номер записи в базе, нумерация с 1 |
| *Возвращаемое значение* | 0 в случае успеха, случаях ошибок ИРБИС – отрицательные значения. FALSE в случае ошибки PHP |
|  |  |
| *См также:* | **IrbisInit** |

|  |  |
| --- | --- |
| **int IrbisReadVersion(resource sp, int mfn)** | **Чтение версии записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Чтение версии записи ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  mfn - номер записи в базе, нумерация с 1 |
| *Возвращаемое значение* | Положительное число в случае успеха, случаях ошибок ИРБИС – отрицательные значения. FALSE в случае ошибки PHP |
|  |  |
| *См также:* | **IrbisInit** |

|  |  |
| --- | --- |
| **string IrbisGetGuid(resource sp, int shelf)** | **GUID записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | GUID записи ИРБИС, находящейся на полке  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | В случае успеха строка GUID, в случае ошибок PHP FALSE |
|  |  |
| *См также:* | **IrbisRecord, IrbisSetGuid** |

|  |  |
| --- | --- |
| **void IrbisSetGuid(resource sp, int shelf, string guid)** | **Смена GUID записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Смена GUID записи ИРБИС, находящейся на полке  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  guid - Новое значение GUID |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | **IrbisRecord, IrbisGetGuid** |

|  |  |
| --- | --- |
| **int IrbisRecordBack(resource sp, int shelf, int mfn, int step)** | **Чтение предыдущей версии записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Чтение предыдущей версии записи ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  mfn - номер записи в базе, нумерация с 1  step количество шагов назад от текущей версии, нумерация с 1 |
| *Возвращаемое значение* | 0 в случае успеха, в остальных случаях коды ошибок ИРБИС (отрицательные значения). FALSE в случае ошибок PHP |
|  |  |
| *См также:* | **IrbisRecord** |

|  |  |
| --- | --- |
| **int IrbisIsDBLocked(resource sp)** | **Проверка блокировки БД ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Проверка блокировки БД ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit |
| *Возвращаемое значение* | 0 в случае успеха, -300 в случае заблокированной базы, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | **IrbisInit** |

|  |  |
| --- | --- |
| **int IrbisIsDeleted(resource sp, int shelf)** | **Проверка удаления записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Проверка удаления записи ИРБИС, загруженной на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | 1 в случае удаленной записи, 0 если запись не удалена, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | **IrbisRecord** |

|  |  |
| --- | --- |
| **int IrbisIsLocked(resource sp, int shelf)** | **Проверка блокировки записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Проверка удаления записи ИРБИС, загруженной на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | 1 в случае заблокированной записи, 0 если запись не заблокирована, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | **IrbisRecord** |

|  |  |
| --- | --- |
| **int IrbisIsRealyLocked(resource sp, int mfn)** | **Чтение флага блокировки записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Проверка блокировки записи по XRF файлу, без загрузки на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  mfn - номер записи в базе, нумерация с 1 |
| *Возвращаемое значение* | 1 в случае заблокированной записи, 0 если запись не заблокирована, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | **IrbisInit** |

|  |  |
| --- | --- |
| **int IrbisIsActualized(resource sp, int shelf)** | **Проверка актуализации записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Проверка актуализации записи, загруженной на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | 1 в случае актуализированной записи, 0 если запись не актуализирована, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | **IrbisRecord** |

|  |  |
| --- | --- |
| **int IrbisIsActualizedFT(resource sp, int shelf)** | **Проверка полнотекстовой актуализации записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Проверка флага полнотекстовой актуализации записи, загруженной на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | 1 в случае актуализированной записи, 0 если запись не актуализирована, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | **IrbisRecord** |

|  |  |
| --- | --- |
| **int IrbisIsRealyActualized(resource sp, int mfn)** | **Чтение флага актуализации записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Проверка актуализации записи по XRF файлу, без загрузки на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  mfn - номер записи в базе, нумерация с 1 |
| *Возвращаемое значение* | 1 в случае актуализированной записи, 0 если запись не актуализирована, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | **IrbisInit** |

|  |  |
| --- | --- |
| **int IrbisSetFullTextActualizedBitTime(resource sp, int mfn, bool FullTextActualized, int seconds)** | **Сохранение флага полнотекстовой актуализации** |

|  |  |
| --- | --- |
| *Описание:* | Сохранение флага полнотекстовой актуализации в базу данных ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  mfn - номер записи в базе, нумерация с 1  **FullTextActualized – флаг актуализации**  **seconds – таймаут для ожидания блокировки базы** |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | **IrbisRecord** |

|  |  |
| --- | --- |
| **void IrbisRecDel(resource sp, int shelf)** | **Удаление записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Установка статуса удаления записи, загруженной на полку без сохранения. Для сохранения записи в базу следует вызвать IrbisRecUpdateTime |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | IrbisRecord, IrbisRecUndelete, IrbisRecUpdateTime |

|  |  |
| --- | --- |
| **void IrbisRecUndelete(resource sp, int shelf)** | **Отмена удаления записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Снятие статуса удаления записи, загруженной на полку без сохранения. Для сохранения записи в базу следует вызвать IrbisRecUpdateTime |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | IrbisRecord, IrbisRecDel, IrbisRecUpdateTime |

|  |  |
| --- | --- |
| **void IrbisRecLock(resource sp, int shelf)** | **Блокировка записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Установка статуса блокировки записи, загруженной на полку без сохранения. Для сохранения записи в базу следует вызвать IrbisRecUpdateTime |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | IrbisRecord, IrbisRecUnlock, IrbisRecUpdateTime |

|  |  |
| --- | --- |
| **void IrbisRecUnlock(resource sp, int shelf)** | **Отмена блокировки записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Снятие статуса блокировки записи, загруженной на полку без сохранения. Для сохранения записи в базу следует вызвать IrbisRecUpdateTime |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | IrbisRecord, IrbisRecLock, IrbisRecUpdateTime |

|  |  |
| --- | --- |
| **void IrbisRecFTActualize (resource sp, int shelf)** | **Установка флага полнотекстовой актуализации записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Установка статуса полнотекстовой актуализации записи, загруженной на полку без сохранения. Для сохранения записи в базу следует вызвать IrbisRecUpdateTime |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | IrbisRecord, IrbisRecFTUnactualize, IrbisRecUpdateTime |

|  |  |
| --- | --- |
| **void IrbisRecFTUnactualize (resource sp, int shelf)** | **Снятие флага полнотекстовой актуализации записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Снятие статуса полнотекстовой актуализации записи, загруженной на полку без сохранения. Для сохранения записи в базу следует вызвать IrbisRecUpdateTime |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | IrbisRecord, IrbisRecFTActualize, IrbisRecUpdateTime |

|  |  |
| --- | --- |
| **Int IrbisRecStatus(resource sp, int shelf, int option)** | **Статус записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Статус записи ИРБИС, загруженной на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  option – 0 – статус записи из MST файла, 1 – статус записи из XRF файла |
| *Возвращаемое значение* | Число больше или равно 0, FALSE в случае ошибок PHP |
|  |  |
| *См также:* | IrbisRecord |

|  |  |
| --- | --- |
| **int IrbisFieldN(resource sp, int shelf, int tag, int occ)** | **Порядковый номер поля** |

|  |  |
| --- | --- |
| *Описание:* | Порядковый номер указанного повторения occ поля с меткой tag в записи, загруженной на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  tag - Метка поля, нумерация с 1  occ - Номер повторения поля, нумерация с 1 |
| *Возвращаемое значение* | Число больше 0 в случае наличия указанного повторения поля, 0 если повторение поля не найдено, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord |

|  |  |
| --- | --- |
| **string IrbisField(resource sp, int shelf, int nf[, string subfields])** | **Значение поля или подполя** |

|  |  |
| --- | --- |
| *Описание:* | Значение поля / подполя записи, загруженной на полку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  nf - Порядковый номер поля в справочнике, нумерация с 1  Для поиска нужного повторения нужного поля используйте IrbisFieldN  subfields - Необязательный пораметр - код подполя в виде ^N  Разрешенные значения – символы с ascii кодами от 33 до 127 |
| *Возвращаемое значение* | строка, в случае ошибок PHP FALSE |
|  |  |
| *См также:* | IrbisRecord, IrbisFieldN |

|  |  |
| --- | --- |
| **int IrbisFldAdd(resource sp, int shelf, int tag, int nf, string field)** | **Добавление поля в запись ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Добавление поля в запись на полке  Запись может быть новая или загружена из базы  Кодировка текста может быть cp1251 и utf8, текст нормализуется до utf8  Добавляется только непустое поле, в случае пустого поля код ошибки -200 |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  tag - Метка поля, нумерация с 1  nf - Порядковый номер поля в справочнике, нумерация с 1  Если nf меньше нуля или больше количества полей в справочнике, поле добавляется в конец справочника  field – значение поля |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisNewRec, IrbisFldRep |

|  |  |
| --- | --- |
| **int IrbisFldRep(resource sp, int shelf, int nf, string field)** | **Замена либо удаление поля в записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Замена либо удаление поля в записи на полке  Запись может быть новая или загружена из базы  Кодировка текста может быть cp1251 и utf8, текст нормализуется до utf8  Если строка пустая, поле будет удалено из справочника, если не пустая - добавлено |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  nf - Порядковый номер поля в справочнике, нумерация с 1  Для поиска нужного повторения нужного поля используйте IrbisFieldN  field – новое значение поля, пустая строка в случае удаления |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisNewRec, IrbisFldAdd |

|  |  |
| --- | --- |
| **int IrbisNFields(resource sp, int shelf)** | **Количество полей в записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Количество полей в записи на полке  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Число больше либо равно 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisNewRec |

|  |  |
| --- | --- |
| **int IrbisNocc(resource sp, int shelf, int tag)** | **Количество повторений поля в записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Количество повторений поля tag в записи на полке shelf  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  tag - Метка поля, нумерация с 1 |
| *Возвращаемое значение* | Число больше либо равно 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisNewRec, IrbisFldAdd |

|  |  |
| --- | --- |
| **int IrbisFldTag(resource sp, int shelf, int nf)** | **Метка поля в записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Метка поля с порядковым номером nf в записи на полке shelf  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  nf - Порядковый номер поля в справочнике, нумерация с 1 |
| *Возвращаемое значение* | Число больше 0 в случае успеха, 0 если поле не найдено, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisNewRec, IrbisNFields |

|  |  |
| --- | --- |
| **int IrbisFldEmpty(resource sp, int shelf)** | **Опустошение записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Опустошение записи на полке  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Число больше 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisNewRec, IrbisNFields |

|  |  |
| --- | --- |
| **int IrbisNewRec(resource sp, int shelf)** | **Создание новой записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Создание новой записи на полке  Предыдущая запись на полке уничтожается |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisInit |

|  |  |
| --- | --- |
| **int IrbisChangeMfn(resource sp, int shelf, int newmfn)** | **Смена номера записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Смена номера записи на полке  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок newmfn - Новый номер записи, нумерация с 1 |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisNewRec |

|  |  |
| --- | --- |
| **int IrbisChangeVersion(resource sp, int shelf, int newversion)** | **Смена версии записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Смена версии записи на полке  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  newversion - Новая версия записи, нумерация с 1 |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisNewRec |

|  |  |
| --- | --- |
| **int IrbisMfn(resource sp, int shelf)** | **Номер записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Номер записи на полке  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Число больше либо равно 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord, IrbisChangeMfn |

|  |  |
| --- | --- |
| **int IrbisVersion(resource sp, int shelf)** | **Версия записи ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Версия записи на полке  Запись может быть новая или загружена из базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок |
| *Возвращаемое значение* | Число больше 0 в случае успеха, 0 если полка не найдена, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecord |

|  |  |
| --- | --- |
| **int IrbisMaxMfn(resource sp)** | **Следующий номер записи в базе** |

|  |  |
| --- | --- |
| *Описание:* | Максимальный номер записи в базе плюс 1  Может быть использован как есть при добавлении новой записи  Нумерация с 1, в пустой базе 1, если в базе одна запись 2 и так далее |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit |
| *Возвращаемое значение* | Число больше 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisInit |

|  |  |
| --- | --- |
| **int IrbisFind(resource sp, string& term)** | **Поиск термина** |

|  |  |
| --- | --- |
| *Описание:* | Поиск термина в словаре базы ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  term - Строка с термином, может быть в кодировке cp1251 и utf8  Текст нормализуется в utf8, фактически найденный термин помещается в эту же строку. Возвращаемый термин будет в кодировке utf8  Ограничение по длине 254 байта.  После вызова IrbisFind можно пользоваться функциями навигации по списку терминов IrbisNxtTerm вперед и IrbisPrevTerm назад.  Полезные коды возврата для работы с функциями IrbisFind, IrbisNxtTerm и IrbisPrevTerm  -202 - термин не найден  -203 - достигнут конец списка  -204 - начало списка |
| *Возвращаемое значение* | 0 в случае, если термин найден, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisNxtTerm, IrbisPrevTerm |

|  |  |
| --- | --- |
| **int IrbisNxtTerm(resource sp, string& term)** | **Переход к следующему термину** |

|  |  |
| --- | --- |
| *Описание:* | Навигация к следующему термину в словаре базы ИРБИС  Применяется после вызова IrbisFind |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  term – Выходной параметр. Строка с термином, который читается из словаря.  Полезные коды возврата для работы с функциями IrbisFind, IrbisNxtTerm и IrbisPrevTerm  -202 - термин не найден  -203 - достигнут конец списка  -204 - начало списка |
| *Возвращаемое значение* | 0 в случае, если термин найден, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisFind, IrbisPrevTerm |

|  |  |
| --- | --- |
| **int IrbisPrevTerm(resource sp, string& term)** | **Переход к предыдущему термину** |

|  |  |
| --- | --- |
| *Описание:* | Навигация к предыдущему термину в словаре базы ИРБИС  Применяется после вызова IrbisFind |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  term – Выходной параметр. Строка с термином, который читается из словаря.  Полезные коды возврата для работы с функциями IrbisFind, IrbisNxtTerm и IrbisPrevTerm  -202 - термин не найден  -203 - достигнут конец списка  -204 - начало списка |
| *Возвращаемое значение* | 0 в случае, если термин найден, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisFind, IrbisNxtTerm |

|  |  |
| --- | --- |
| **int IrbisNPosts(resource sp)** | **Количество ссылок на термин** |

|  |  |
| --- | --- |
| *Описание:* | Количество ссылок на термин  Применяется после IrbisFind, IrbisNxtTerm и IrbisPrevTerm  Сразу после вызова IrbisFind, IrbisNxtTerm и IrbisPrevTerm указатель встает на первую ссылку текущего термина  Навигация по списку ссылок на термин возможна только вперед с помощью IrbisNxtPost |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit |
| *Возвращаемое значение* | Число больше либо равно 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения).  Значение 0 означает, что записи были удалены, в этом случае просто обнуляется количество ссылок |
|  |  |
| *См также:* | IrbisFind, IrbisNxtTerm, IrbisPrevTerm |

|  |  |
| --- | --- |
| **int IrbisPosting(resource sp, int opt)** | **Поле записи со ссылкой на термин** |

|  |  |
| --- | --- |
| *Описание:* | Количество ссылок на термин  Применяется после IrbisFind, IrbisNxtTerm и IrbisPrevTerm  Сразу после вызова IrbisFind, IrbisNxtTerm и IrbisPrevTerm указатель встает на первую ссылку текущего термина  Навигация по списку ссылок на термин возможна только вперед с помощью IrbisNxtPost |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  opt - Номер поля записи со ссылкой  Возможные значения  Для понимания см документацию по таблицам выбора полей ИРБИС  Для ИРБИС до 2017 версии включительно  1 - Номер записи  2 - Метка строки в ТВП  3 - Номер повторения в тексте результата выполнения формата строки таблицы выбора полей (ТВП)  (разбиение на повторения делается по символу %)  4 - Номер слова в тексте результата выполнения формата, зависит от метода индексирования  Для версии ИРБИС 2018 + и более поздних  1 - номер записи  2 - номер записи 1-й страницы текста  3 - номер записи бибописания текста в ЭК  4 - номер слова на странице  5 - число слов в предложении  6 - номер страницы  7 - номер предложения  8 - номер слова в предложении |
| *Возвращаемое значение* | Число больше либо равно 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения). |
|  |  |
| *См также:* | IrbisFind, IrbisNxtTerm, IrbisPrevTerm, IrbisNPosts |

|  |  |
| --- | --- |
| **int IrbisNxtPost(resource sp)** | **Навигация к следующей ссылке на термин** |

|  |  |
| --- | --- |
| *Описание:* | Навигация к следующей ссылке на термин  Применяется после IrbisFind, IrbisNxtTerm и IrbisPrevTerm |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisFind, IrbisNxtTerm, IrbisPrevTerm, IrbisPosting |

|  |  |
| --- | --- |
| **int IrbisDBEmptyTime(resource sp, int seconds)** | **Опустошение базы данных ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Опустошение базы ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  seconds - Время ожидания монопольной блокировки базы (по мьютексам) в секундах |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisInit |

|  |  |
| --- | --- |
| **int IrbisLockDBTime(resource sp, int seconds)** | **Блокирование базы данных ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Блокирование базы ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  seconds - Время ожидания монопольной блокировки базы (по мьютексам) в секундах |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisInit |

|  |  |
| --- | --- |
| **int IrbisUnLockDBTime(resource sp, int seconds)** | **Разблокирование базы данных ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Разблокирование базы ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  seconds - Время ожидания монопольной блокировки базы (по мьютексам) в секундах |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisInit |

|  |  |
| --- | --- |
| **int IrbisRecUpdateTime(resource sp, int shelf, bool KeepLock, bool Updif, int seconds, int& result\_update, int& result\_updif)** | **Сохранение записи в базу данных ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Сохранение записи в базу данных ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  seconds - Время ожидания монопольной блокировки базы (по мьютексам) в секундах  KeepLock - Оставить запись заблокированной после обновления  Updif - Обновить или нет словарь  Обновить отдельно словарь можно с помощью функции IrbisRecIfUpdateTime  seconds - Время ожидания монопольной блокировки базы (по мьютексам) в секундах  result\_update - Выходной параметр для результата обновления записи  result\_updif - Выходной параметр для результата обноления словаря |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecIfUpdateTime |

|  |  |
| --- | --- |
| **int IrbisRecIfUpdateTime(resource sp, int shelf, int mfn, int seconds)** | **Актуализация записи** |

|  |  |
| --- | --- |
| *Описание:* | Обновление словаря по записи с указанным номером (актуализация) |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  Запись будет загружена на полку, предварительно загружать запись не нужно  mfn - Номер записи, нумерация с 1  seconds - Время ожидания монопольной блокировки базы (по мьютексам) в секундах  Возвращаемое значение 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisRecUpdateTime |

|  |  |
| --- | --- |
| **int IrbisRecLockTime(resource sp, int shelf, int mfn, int seconds)** | **Чтение записи с блокировкой** |

|  |  |
| --- | --- |
| *Описание:* | Чтение записи с блокировкой |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  Запись будет загружена на полку, предварительно загружать запись не нужно  mfn - Номер записи, нумерация с 1  seconds - Время ожидания монопольной блокировки базы (по мьютексам) в секундах |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisInit, IrbisRecord |

|  |  |
| --- | --- |
| **int IrbisRecUnLockTime(resource sp, int mfn, int seconds)** | **Разблокирование записи** |

|  |  |
| --- | --- |
| *Описание:* | Разблокирование записи без чтения (только в XRF) |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  mfn - Номер записи, нумерация с 1  seconds - Время ожидания монопольной блокировки базы (по мьютексам) в секундах |
| *Возвращаемое значение* | 0 в случае успеха, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | IrbisInit, IrbisRecLockTime |

|  |  |
| --- | --- |
| **string IrbisDLLVersion()** | **Версия модуля** |

|  |  |
| --- | --- |
| *Описание:* | Версия модуля |
| *Возвращаемое значение* | Строка с версией модуля |
|  |  |

|  |  |
| --- | --- |
| **int Irbis\_InitPFT(resource sp, string Line)** | **Загрузка формата в виде текста или из файла** |

|  |  |
| --- | --- |
| *Описание:* | Загрузка формата в виде текста или из файла  Для загрузки из файла первым символом должен быть символ @  Если формат представлен в тексте, то для указания кодировки текста следует использовать символ !  Если первый символ ! то формат понимается как текст в кодировке utf8  Если первый символ не ! то формат перекодируется из cp1251 в utf8  В случае с форматером xpft формат компилируется и в случае ошибки будет код ошибки больше нуля  Также описание ошибки будет выведено как предупреждение PHP  В случае с классическим форматером ошибки в формате определяются на этапе выполнения в функции Irbis\_Format  По умолчанию включен классический форматер, для переключения форматеров используйте функцию irbis\_enable\_xpft.  Если текст в кодировке utf8, то следует использовать в начале символ !  В противном случае текст перекодируется из cp1251 в utf8  Также следует иметь в виду, что символ ! в любом месте, кроме первого символа - это escape литерал ISIS  Старая конструкция для вывода escape последовательностей в текстовых терминалах  В документации по ИРБИС и ISIS этого нет  Если интерпретатор в тексте встречает символ !, то следующий за ним символ считается разделителем  Интерпретатор находит следующее вхождение этого символа и текст между ними выводит без обработки, предваряя вывод  символом с ASCII кодом 27  Поведение похоже на безусловный литерал, только в начале добавляется ASCII символ 27  Пример (отодвинем ! от первой позиции в строке с помощью разделителя ,)  ,!'Я escape литерал' |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  Line - имя файла формата (с символом @ в начале), либо текст |
| *Возвращаемое значение* | 0 в случае успеха, коды больше нуля - ошибки синтаксиса (xpft), в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | Irbis\_enable\_xpft, Irbis\_Format |

|  |  |
| --- | --- |
| **int Irbis\_Format(resource sp, int Curr\_Shelf, int Alt\_Shelf, string &fmt)** | **Выполнение формата** |

|  |  |
| --- | --- |
| *Описание:* | Выполнение формата, загруженного с помощью Irbis\_InitPFT |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  Curr\_Shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  Расформатируемая запись должна находиться на полке Curr\_Shelf  Alt\_Shelf - Номер полки, куда будут загружаться другие записи, если используется функция REF  fmt - Выходная переменная с результатом форматирования |
| *Возвращаемое значение* | 0 в случае успеха, коды больше нуля - ошибки синтаксиса, в случае ошибок PHP FALSE, в остальных случаях коды ошибок ИРБИС (отрицательные значения) |
|  |  |
| *См также:* | Irbis\_InitPFT |

|  |  |
| --- | --- |
| **string IrbisDbName(resource sp)** | **Имя текущей базы** |

|  |  |
| --- | --- |
| *Описание:* | Имя текущей базы |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit |
| *Возвращаемое значение* | Возвращаемое значение - строка с именем базы |
|  |  |
| *См также:* | IrbisInit |

|  |  |
| --- | --- |
| **array IrbisFTSearch(resource sp, string request, string request\_search, string request\_scan, bool morphology, string prefix, string ftprefix, int maxwordsdistance)** | **Полнотекстовый поиск** |

|  |  |
| --- | --- |
| *Описание:* | Полнотекстовый поиск ИРБИС  Работает аналогично unifor +3F  Для баз 2017 версии по сути представляет собой обертку для функции  array IrbisRangeSearchContext\_In(resource sp, string request, string request\_in, string request\_search, string request\_scan, bool morphology, string prefix, int maxWordsDistanse, array context, int maxcountresult)  только параметры context и maxcountresult задаются по умолчанию: context пусто и maxcountresult 1000000  Если эти параметры нужны, для баз 2018 версии лучше использовать напрямую IrbisRangeSearchContext\_In |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  request - поисковый запрос на естественном языке  request\_search - поисковая строка на языке ISIS  request\_scan - формулировка последовательного поиска  morphology - учитывать морфологию слова или производить усечение?  prefix - префикс слов в ЭК  ftprefix - префикс слов в ПБД  maxwordsdistanse - максимальное расстояние между словами, если 0 то не работает (2018 +, в 2017 не используется) |
| *Возвращаемое значение* | в случае успеха ассоциативный массив, в случае ошибок PHP FALSE  Ассоциативный массив с результатами содержит следующие ключи  mfnlist - результатами поиска список MFN, одномерный массив с числами  requestlist - для версии 2017 список выполненных запросов с ненулевыми ответами в порядке очередности  ассоциативный массив ТЕРМИН => количество mfn, в версии 2018+ пусто  mfnpages - для версии 2018+ результат поиска - соответствующий MfnList релевантный список записей (страниц) PDF, массив, каждый элемент  которого является массивом из двух элементов, в 2017 версии пусто  0 - MFN записи, число  1 - список страниц, массив чисел  mfntagsnew - для версии 2018+ массив, ключ - MFN записи в ЭК, значение MFN записи в ПБД содержащий первую релевантную страницу |
|  |  |
| *См также:* | IrbisRangeSearchContext\_In, IrbisFreeSearch |

|  |  |
| --- | --- |
| **array IrbisRangeSearchContext\_In(resource sp, string request, string request\_in, string request\_search, string request\_scan, bool morphology, string prefix, int maxWordsDistanse, array context, int maxcountresult)** | Структурированный поиск и поиск в найденном |

|  |  |
| --- | --- |
| *Описание:* | Структурированный поиск и поиск в найденном (если задано request\_in) |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  request - поисковый запрос на естественном языке  request\_in - искать эти слова в найденном по request  request\_search - поисковая строка на языке ISIS  request\_scan - формулировка последовательного поиска  morphology - учитывать морфологию слова или производить усечение?  prefix - префикс слов в ЭК  maxWordsDistanse - максимальное расстояние между словами, если 0 то не работает  context - набор тематических индексов  maxcountresult - максимальное число ответов |
| *Возвращаемое значение* | в случае успеха ассоциативный массив, в случае ошибок PHP FALSE  Ассоциативный массив с результатами содержит следующие ключи  requestlist - список выполненных запросов с ненулевыми ответами в порядке очередности  ассоциативный массив ТЕРМИН => количество mfn  mfnlist - результатами поиска список MFN, одномерный массив с числами |
|  |  |
| *См также:* | IrbisFTSearch, IrbisFreeSearch |

|  |  |
| --- | --- |
| **array IrbisSearchCells(resource sp, array mfnlist, int cellcount)** | **Поиск по фасетам** |

|  |  |
| --- | --- |
| *Описание:* | Поиск по фасетам  Официальный комментарий:  считает фасеты на основе файла DBNAME.CELLS  результат список фасетов CellsList с количеством ссылок в найденных записях  Если по человечески:  сортировка списка mfn по релевантности |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  mfnlist - список mfn от предыдущего поиска  count - количество результатов |
| *Возвращаемое значение* | в случае успеха ассоциативный массив, в случае ошибок PHP FALSE |
|  |  |
| *См также:* | IrbisFTSearch, IrbisFreeSearch, IrbisRangeSearchContext\_In |

|  |  |
| --- | --- |
| **array IrbisFreeSearch(resource sp, string request, bool block, bool updownResult, bool morphology = FALSE, string prefix = '')** | **Поиск по поисковому выражению на языке запросов ISIS** |

|  |  |
| --- | --- |
| *Описание:* | Поиск по поисковому выражению на языке запросов ISIS.  Опционально с блокировкой базы, можно указать, чтобы результаты были в обратном порядке |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  request - поисковая строка на языке ISIS  block - блокирование базы на момент поиска  updownResult - результаты в обратном порядке  morphology – использовать морфологию  prefix – префикс морфологии |
| *Возвращаемое значение* | в случае успеха одномерный массив MFN, в случае ошибок PHP FALSE |
|  |  |
| *См также:* | IrbisFTSearch, IrbisRangeSearchContext\_In |

|  |  |
| --- | --- |
| **int IrbisNewDB(resource ctx, string name, string description, bool reader, bool readerweb, string & msg)** | **Создание новой базы данных** |

|  |  |
| --- | --- |
| *Описание:* | Создание новой базы ИРБИС  Формат файлов новой БД определяется по эталонной базе данных  Имя эталонной БД в ini файле секция Main параметр EtalonDBN, по умолчанию IBIS. Требует опции SDK в лицензионном ключе. Без этого не работает. |
| *Параметры:* | ctx – контекст, созданный IrbisAppInit  name - имя базы  description - описание  Текст может быть в кодировке cp1251 и utf8, при при записи в файл нормализуется в cp1251  reader - база доступна читателю  readerweb - база доступна в WEB ИРБИС  msg - выходной параметр для сообщения об ошибке |
| *Возвращаемое значение* | 0 в случае успеха, отрицательные значения - ошибки ИРБИС, FALSE в случае ошибок PHP |
|  |  |
| *См также:* | IrbisAppInit, IrbisDeleteDB |

|  |  |
| --- | --- |
| **int IrbisDeleteDB(resource ctx, string name, string & msg)** | **Удаление базы данных** |

|  |  |
| --- | --- |
| *Описание:* | Удаление базы ИРБИС.  Требует опции SDK в лицензионном ключе. Без этого не работает. |
| *Параметры:* | ctx – контекст, созданный IrbisAppInit  name - имя базы  msg - выходной параметр для сообщения об ошибке |
| *Возвращаемое значение* | 0 в случае успеха, отрицательные значения - ошибки ИРБИС, FALSE в случае ошибок PHP |
|  |  |
| *См также:* | IrbisNewDB |

|  |  |
| --- | --- |
| **void IrbisActualizeFullText(resource ctx, int mfn, bool mfn\_del, string filename, string dbname, string guid)** | **Актуализация полного текста** |

|  |  |
| --- | --- |
| *Описание:* | Актуализация записи в полнотекстовой базе ИРБИС. |
| *Параметры:* | ctx – контекст, созданный IrbisAppInit  mfn – MFN записи в полнотекстовой базе  mfn\_del – признак того, что полный текст отвязан от документа в базе электронного каталога  filename - имя файла pdf  dbname – имя базы электронного каталога  guid – GUID записи электронного каталога в базе dbname |
| *Возвращаемое значение* | Функция не возвращает значение |
|  |  |

|  |  |
| --- | --- |
| **string irbis\_server\_ini(resource ctx, string section, string key, string defaultValue)** | **Чтение параметра из серверного ini файла** |

|  |  |
| --- | --- |
| *Описание:* | Чтение параметра из серверного ini файла |
| *Параметры:* | ctx - контекст приложения, открытый функцией IrbisAppInit  section - имя секции  key - имя параметра  defaultValue - значение по умолчанию |
| *Возвращаемое значение* | При успешном выполнении - строка со значением параметра, FALSE в случае ошибок PHP |
|  |  |
| *См также:* | IrbisAppInit,irbis\_server\_ini\_put |

|  |  |
| --- | --- |
| **void irbis\_server\_ini\_put(resource ctx, string section, string key, string value)** | **Установка параметра серверного ini файла в памяти** |

|  |  |
| --- | --- |
| *Описание:* | Установка параметра серверного ini файла.  Параметр устанавливается в памяти, в файл не записывается |
| *Параметры:* | ctx - контекст приложения, открытый функцией IrbisAppInit  section - имя секции  key - имя параметра  value - значение |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | IrbisAppInit,irbis\_server\_ini |

|  |  |
| --- | --- |
| **void irbis\_load\_user\_ini(resource ctx, string path)** | **Загрузка пользовательского ini файла на место серверного** |

|  |  |
| --- | --- |
| *Описание:* | Загрузка пользовательского ini файла вместо серверного, который был указан при создании контекста IrbisAppInit |
| *Параметры:* | ctx - контекст приложения, открытый функцией IrbisAppInit  path – путь к ini файлу |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | IrbisAppInit,irbis\_server\_ini |

|  |  |
| --- | --- |
| **string irbis\_flatten\_ini(string path)** | **Чтение ini файла вместе со вложенными в плоский список** |

|  |  |
| --- | --- |
| *Описание:* | Чтение ini файла вместе со вложенными в единый плоский список. При слиянии опции, которые уже есть в файле верхнего уровня, на такие же опции из вложенных файлов не заменяются. |
| *Параметры:* | path – путь к ini файлу |
| *Возвращаемое значение* | Текст ini файла |
|  |  |

|  |  |
| --- | --- |
| **resource irbis\_ini\_load(string path)** | **Загрузка ini файла ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Загрузка ini файла ИРБИС для редактирования |
| *Параметры:* | path – путь к ini файлу |
| *Возвращаемое значение* | Идентификатор ресурса в случае успеха, иначе FALSE |
|  |  |
| *См также:* | irbis\_ini\_save, irbis\_ini\_close, irbis\_ini\_get, irbis\_ini\_put |

|  |  |
| --- | --- |
| **void irbis\_ini\_close(resource ctx)** | **Закрытие ini файла ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Закрытие ini файла ИРБИС, открытого функцией irbis\_ini\_load |
| *Параметры:* | ctx – идентификатор ресурса, полученный при открытии файла функцией irbis\_ini\_load |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | irbis\_ini\_load, irbis\_ini\_save, irbis\_ini\_get, irbis\_ini\_put |

|  |  |
| --- | --- |
| **int irbis\_ini\_save(resource ctx, int seconds)** | **Сохранение ini файла ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Сохранение ini файла ИРБИС, открытого функцией irbis\_ini\_load |
| *Параметры:* | ctx – идентификатор ресурса, полученный при открытии файла функцией irbis\_ini\_load  seconds - Время ожидания очереди на сохранение (по мьютексам) в секундах |
| *Возвращаемое значение* | 0 в случае успеха, отрицательные значения - ошибки ИРБИС, FALSE в случае ошибок PHP |
|  |  |
| *См также:* | irbis\_ini\_load, irbis\_ini\_close, irbis\_ini\_get, irbis\_ini\_put |

|  |  |
| --- | --- |
| **string irbis\_ini\_get(resource ctx, string section, string key, string defaultValue)** | **Чтение параметра из ini файла ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Чтение параметра из ini файла ИРБИС, открытого функцией irbis\_ini\_load |
| *Параметры:* | ctx – идентификатор ресурса, полученный при открытии файла функцией irbis\_ini\_load  section – имя секции  key – имя параметра  defaultValue – значение по умолчанию |
| *Возвращаемое значение* | Строка со значением параметра, FALSE в случае ошибок PHP |
|  |  |
| *См также:* | irbis\_ini\_load, irbis\_ini\_close, irbis\_ini\_save, irbis\_ini\_put |

|  |  |
| --- | --- |
| **void irbis\_ini\_put(resource ctx, string section, string key, string value)** | **Установка параметра в ini файле ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Установка параметра в ini файле ИРБИС, открытом функцией irbis\_ini\_load |
| *Параметры:* | ctx – идентификатор ресурса, полученный при открытии файла функцией irbis\_ini\_load  section – имя секции  key – имя параметра  value – значение |
| *Возвращаемое значение* | Функция не возвращает значения |
|  |  |
| *См также:* | irbis\_ini\_load, irbis\_ini\_close, irbis\_ini\_save, irbis\_ini\_get |

|  |  |
| --- | --- |
| **string IrbisUnpackSearchExpression(string expr)** | **Разбиение запроса на языке ISIS в массив** |

|  |  |
| --- | --- |
| *Описание:* | Разбиение запроса на языке ISIS в массив |
| *Параметры:* | expr - поисковая строка на языке ISIS |
| *Возвращаемое значение* | При успешном выполнении - ассоциативный массив, FALSE в случае ошибок PHP  Структура массива  Ключ - термин, значение номер оператора по списку +, \*, ^, G, F и так далее |
|  |  |

|  |  |
| --- | --- |
| **array IrbisTextToWords\_Range(resource sp, string request)** | **Разбиение фразы на слова с ранжированием** |

|  |  |
| --- | --- |
| *Описание:* | Разбиение фразы на слова с ранжированием (якобы) |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  request - фраза на естественном языке |
| *Возвращаемое значение* | При успешном выполнении - ассоциативный массив, FALSE в случае ошибок PHP  Структура массива  Ключ - слово, значение номер слова в фразе |
| *См также* | IrbisInit |
|  |  |

|  |  |
| --- | --- |
| **array IrbisTextToWords(resource sp, string request, bool deflex)** | **Разбиение фразы на слова** |

|  |  |
| --- | --- |
| *Описание:* | Разбиение фразы на слова  опционально можно включить отбрасывание окончаний слов |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  request - фраза на естественном языке  deflex - отбрасывать или нет окончания слов |
| *Возвращаемое значение* | При успешном выполнении - ассоциативный массив, FALSE в случае ошибок PHP  Структура массива  Ключ - слово, значение номер слова в фразе |
| *См также* | IrbisInit |
|  |  |

|  |  |
| --- | --- |
| **array IrbisExtractTermsFromText(resource sp, resource morph, string text, bool deflex)** | **Разбиение фразы на слова с морфологией** |

|  |  |
| --- | --- |
| *Описание:* | Разбиение фразы на слова с морфологией  опционально можно включить отбрасывание окончаний слов |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  mofph - База морфологии (обычно MORPH), открытая функцией IrbisInit  text - фраза на естественном языке  deflex - отбрасывать или нет окончания слов |
| *Возвращаемое значение* | При успешном выполнении - ассоциативный массив, FALSE в случае ошибок PHP |
| *См также* | IrbisInit |
|  |  |

|  |  |
| --- | --- |
| **int IrbisFlexMN(string term)** | **Определение длины окончания слова для русского языка** |

|  |  |
| --- | --- |
| *Описание:* | Определение длины окончания слова для русского языка  Строка должна быть в utf-8 |
| *Параметры:* | term - слово |
| *Возвращаемое значение* | Длина окончания или 0 |
|  |  |

|  |  |
| --- | --- |
| **int IrbisPDFGetNumPages(resource ctx, string pdfname, string pasword)** | **Количество страниц в PDF документе** |

|  |  |
| --- | --- |
| *Описание:* | Количество страниц в PDF документе |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  pdfname - Путь к pdf файлу  password - Пароль |
| *Возвращаемое значение* | При успешном выполнении - количество страниц в файле, FALSE в случае ошибок PHP |
| *Примечание* | Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами |
| *См также* | IrbisAppInit, IrbisExtractPDFPageGroup |
|  |  |

|  |  |
| --- | --- |
| **int IrbisExtractPDFPageGroup(resource ctx, string pdfname, string password, int page, string pdfpagename, int extractpagecount)** | **Извлечение страниц из PDF файла** |

|  |  |
| --- | --- |
| *Описание:* | Извлечение указанного количества страниц из PDF файла  Если выходной файл существует, ничего не делает.  Добавляет номер страницы к имени каждого постраничного файла  Например, c:\путь\1файл.pdf, c:\путь\2файл.pdf и т.д. |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  pdfname - Путь к pdf файлу  password - Пароль  page - Номер первой страницы  pdfpagename - Имя выходного файла  extractpagecount - Количество страниц для извлечения |
| *Возвращаемое значение* | При успешном выполнении - 0, при ошибках работы с PDF -1, FALSE в случае ошибок PHP |
| *Примечание* | Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами |
| *См также* | IrbisAppInit, IrbisPDFGetNumPages |
|  |  |

|  |  |
| --- | --- |
| **array IrbisGlobal(resource sp, int shelf, array gbl)** | **Глобальная корректировка** |

|  |  |
| --- | --- |
| *Описание:* | Выполнение глобальной корректировки ИРБИС  Выполняет модификацию текущей записи ИРБИС, при наличии команд по работе с другими записями  возвращается список ресурсов БД с ними.  Изменения требуется сохранять вручную с помощью IrbisRecUpdateTime  Наличие непустого сообщения в err обозначает ошибку |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  gbl - Массив строк глобальной корректировки, может быть предварительно загружен из файла или сформирован в коде |
| *Возвращаемое значение* | При успешном выполнении - ассоциативный массив, FALSE в случае ошибок PHP  Структура массива  err - сообщение об ошибке  log - вывод команд PUTLOG  splist - список ресурсов БД, аналогичных вызову IrbisInit  Данный список формируется командами создать создания, удаления записи.  Если в нем есть элементы, их нужно сохранять вручную с помощью IrbisRecUpdateTime, в конце закрыть с помощью IrbisClose |
| *См также* | IrbisInit, IrbisRecUpdateTime |
|  |  |

|  |  |
| --- | --- |
| **array IrbisXrfInfo(resource sp, bool logdel, bool physdel, bool notact, bool notactft, bool locked)** | **Опрос базы данных ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Опрос базы данных ИРБИС по XRF файлу. Выполняется поиск удаленных, физически удаленных, неактуализированных или заблокированных записей |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  logdel – выводить список MFN логически удаленных записей  physdel – выводить список MFN физически удаленных записей  notact – выводить список MFN неактуализированных записей  notactft – выводить список MFN записей с неактуализированным полным текстом  locked – выводить список MFN заблокированных записей |
| *Возвращаемое значение* | При успешном выполнении - ассоциативный массив, FALSE в случае ошибок PHP  Структура массива  logdel – список MFN логически удаленных записей  physdel – список MFN физически удаленных записей  notact – список MFN неактуализированных записей  notactft – список MFN записей с неактуализированным полным текстом  locked – список MFN заблокированных |
| *См также* | IrbisInit |
|  |  |

|  |  |
| --- | --- |
| **string webirbis\_keyencode(string barcode)** | **Кодирование ключа безопасности WEB ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Кодирование ключа безопасности WEB ИРБИС |
| *Параметры:* | sp - База ИРБИС, открытая функцией IrbisInit  shelf - Номер полки (буфера записи), нумерация с нуля, доступно 7 полок  gbl - Массив строк глобальной корректировки, может быть предварительно загружен из файла или сформирован в коде |
| *Возвращаемое значение* | barcode - Идентификатор читателя |
| *См также* | webirbis\_keydecode |
|  |  |

|  |  |
| --- | --- |
| **bool webirbis\_keydecode(string key, string& barcode)** | **Декодирование ключа безопасности WEB ИРБИС** |

|  |  |
| --- | --- |
| *Описание:* | Кодирование ключа безопасности WEB ИРБИС |
| *Параметры:* | key - ключ безопасности WEB ИРБИС  barcode - выходной параметр, в него помещается штрихкод читателя  Возвращаемое значение  TRUE если ключ успешно раскодирован и действующий, иначе FALSE |
| *Возвращаемое значение* | barcode - Идентификатор читателя |
| *См также* | webirbis\_keydecode |
|  |  |

|  |  |
| --- | --- |
| **resource webirbis\_pdf\_doc\_open(resource ctx, string filename, string password)** | **Открытие pdf документа** |

|  |  |
| --- | --- |
| *Описание:* | Открытие pdf документа |
| *Параметры:* | ctx - Контекст, созданный функцией IrbisAppInit  filename - Путь к файлу .pdf  password - Пароль |
| *Возвращаемое значение* | Идентификатор ресурса, либо FALSE в случае ошибок |
| *Примечание* | Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами |
| *См также* | webirbis\_pdf\_doc\_close |
|  |  |

|  |  |
| --- | --- |
| **void webirbis\_pdf\_doc\_close(resource doc)** | **Закрытие pdf документа** |

|  |  |
| --- | --- |
| *Описание:* | Открытие pdf документа |
| *Параметры:* | doc - Идентификатор ресурса, созданный функцией webirbis\_pdf\_doc\_open  filename - Путь к файлу .pdf  password - Пароль |
| *Возвращаемое значение* | Функция не возвращает значения |
| *Примечание* | Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами |
| *См также* | webirbis\_pdf\_doc\_open |
|  |  |

|  |  |
| --- | --- |
| **int webirbis\_pdf\_page\_count(resource doc)** | **Количество страниц в pdf документе** |

|  |  |
| --- | --- |
| *Описание:* | Открытие pdf документа |
| *Параметры:* | doc - Идентификатор ресурса, созданный функцией webirbis\_pdf\_doc\_open |
| *Возвращаемое значение* | число больше нуля, или 0 в случае ошибок |
| *Примечание* | Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами |
| *См также* | webirbis\_pdf\_doc\_open, webirbis\_pdf\_doc\_close |
|  |  |

|  |  |
| --- | --- |
| **string webirbis\_pdf\_page\_pdf(resource doc, int pagenum, int pagecount, array words)** | **Порция PDF документа в виде PDF** |

|  |  |
| --- | --- |
| *Описание:* | Возврат порции документа в виде PDF с опциональной подсветкой слов |
| *Параметры:* | doc - Идентификатор ресурса, созданный функцией webirbis\_pdf\_doc\_open  pagenum - Номер первой страницы, нумерация с 1  pagecount - Количество страниц  words - Одномерный массив слов для подсветки |
| *Возвращаемое значение* | Бинарная строка, которую можно сохранить в файл или выдать в выходной поток, либо FALSE в случае ошибок |
| *Примечание* | Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами |
| *См также* | webirbis\_pdf\_doc\_open, webirbis\_pdf\_doc\_close |
|  |  |

|  |  |
| --- | --- |
| **string webirbis\_pdf\_page\_png(resource doc, int pagenum, int dpi, array words)** | **Страница PDF документа в виде PNG** |

|  |  |
| --- | --- |
| *Описание:* | Возврат страницы в PNG с опциональной подсветкой слов |
| *Параметры:* | doc - Идентификатор ресурса, созданный функцией webirbis\_pdf\_doc\_open  pagenum - Номер страницы, нумерация с 1  dpi – Разрешение  words - Одномерный массив слов для подсветки |
| *Возвращаемое значение* | Бинарная строка, которую можно сохранить в файл или выдать в выходной поток, либо FALSE в случае ошибок |
| *Примечание* | На самом деле, в .PNG картинках DPI неприменимо, в данном случае от DPI рассчитывается масштабирование изображения  При значении параметра 72 масштабирования нет, при других значениях изображение делается больше или меньше.  Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами. |
| *См также* | webirbis\_pdf\_doc\_open, webirbis\_pdf\_doc\_close |
|  |  |

|  |  |
| --- | --- |
| **string webirbis\_pdf\_page\_text(resource doc, int pagenum, int pagecount)** | **Порция PDF документа в виде текста** |

|  |  |
| --- | --- |
| *Описание:* | Возврат порции документа в виде текста |
| *Параметры:* | doc - Идентификатор ресурса, созданный функцией webirbis\_pdf\_doc\_open  pagenum - Номер первой страницы, нумерация с 1  pagecount - Количество страниц |
| *Возвращаемое значение* | Текст одной или нескольких страниц, разделенных переводом каретки, либо FALSE в случае ошибок |
| *Примечание* | На самом деле, в .PNG картинках DPI неприменимо, в данном случае от DPI рассчитывается масштабирование изображения  При значении параметра 72 масштабирования нет, при других значениях изображение делается больше или меньше.  Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами. |
| *См также* | webirbis\_pdf\_doc\_open, webirbis\_pdf\_doc\_close |
|  |  |

|  |  |
| --- | --- |
| **string webirbis\_pdf\_page\_words(resource doc, int pagenum)** | **Слова с координатами из страницыPDF** |

|  |  |
| --- | --- |
| *Описание:* | Возврат слов с координатами в виде текста для указанной страницы  Координаты рассчитываются из разрешения 72 dpi  Для рисования средствами PHP под другое разрешение их нужно масштабировать вручную |
| *Параметры:* | doc - Идентификатор ресурса, созданный функцией webirbis\_pdf\_doc\_open  pagenum - Номер страницы, нумерация с 1 |
| *Возвращаемое значение* | Строка слов с координатами, разделитель перевод каретки |
| *Примечание* | Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами. |
| *См также* | webirbis\_pdf\_doc\_open, webirbis\_pdf\_doc\_close |
|  |  |

|  |  |
| --- | --- |
| **array webirbis\_pdf\_find\_words(string text, array words)** | **Поиск слов с координатами** |

|  |  |
| --- | --- |
| *Описание:* | Поиск слов с координатами в тексте, извлеченном функцией webirbis\_pdf\_page\_words |
| *Параметры:* | text - Слова с координатами, извлеченные функцией. webirbis\_pdf\_page\_words  words - Одномерный массив слов |
| *Возвращаемое значение* | Массив координат структура array ( array( $x0, $y0, $x1, $y1) ... ) |
| *Примечание* | Поиск производится по точному соответствию слова, чтобы искать по началу слова следует в конце слова добавить символ $  Все функции по работе с PDF рекомендуется использовать в отдельном процессе во избежание проблем со сторонними компонентами. |
| *См также* | webirbis\_pdf\_page\_words |
|  |  |