

## Опыт разработки базового ПО для реализации библиотечных проектов на основе САБ ИРБИС64 в среде .NET

### The experience of developing basic .NET software for IRBIS64 LAS – based library projects

*И. С. Батрак*

*Библиотечно-издательский комплекс Сибирского федерального университета,  
Красноярск, Россия*

*Ivan Batrak*

*Siberian Federal University Library and Publishing Complex,  
Krasnoyarsk, Russia*

В докладе рассматриваются вопросы разработки базовых компонентов кроссплатформенного программного обеспечения для реализации библиотечных проектов на базе САБ ИРБИС64 в среде .NET 4.0 на примере реализации WEB приложения ИРБИС64 для ASP.NET. Рассмотрен опыт разработки собственной реализации важнейших компонентов САБ ИРБИС64 на основе архитектуры J-ISIS: языка форматирования PFT, альтернативной библиотеки функций unifor и umarci, языка глобальных корректировок, языка поисковых запросов CDS ISIS.

**Ключевые слова:** язык форматирования, unifor, Web-ИРБИС64.

The development of the basic components for cross-platform IRBIS LAS based library software using .NET 4.0 is described in the paper. IRBIS web gateway ASP.NET implementation is used as an example. The experience of developing most important IRBIS LAS features using J-ISIS architecture (such as print format language PFT, alternative unifor and umarci functions library, global correction language, CDS ISIS query language) is also described.

**Keywords:** print format language, unifor, web irbis.

#### **История проекта**

В рамках реализации проектов, связанных с библиотечными сервисами возникла потребность наладить взаимодействие личного кабинета читателя СФУ с внешними приложениями. Личный кабинет читателя СФУ построен на базе WEB приложения ИРБИС64 и не позволял каким-либо образом наладить переходы из внешнего мира к разделам, требующим авторизации.

Система авторизации WEB приложения ИРБИС64 (далее Web-ИРБИС64) реализована с помощью добавления зашифрованного ключа во все ссылки и формы. Сессии и куки не поддерживаются. Единственная возможность авторизовать внешнего пользователя без отображения формы авторизации Web-ИРБИС64 – это включить идентификатор читателя в открытом виде в ссылку на раздел личного кабинета. Формирование подобных ссылок нежелательно с точки зрения защиты персональных данных читателя.

Основным языком разработки WEB сайтов на платформе Web-ИРБИС64 является язык форматирования ИРБИС. Это специализированный язык программирования, основан на языке PFT (print format language), разработанном для приложения CDS ISIS, предшественника ИРБИС. Представляет собой интерпретируемый сценарий, входными данными, для которого является запись базы данных. На выходе такой сценарий генерирует текст. Используется как для создания печатных форм, так и для формирования поискового индекса баз данных.

Сложность в модернизации личного кабинета читателя СФУ состоит в том, что интерпретатор сценариев Web-ИРБИС64 является частью приложения Web-ИРБИС64 и какие-либо альтернативные реализации отсутствуют. Интерпретатор язык форматирования ИРБИС является частью базовой библиотеки САБ ИРБИС64 (далее ИРБИС) – irbis64.dll. Использование библиотеки irbis64.dll вне приложений, написанных на Delphi, затруднено. Платформа Web-ИРБИС64 не позволяет каким-либо образом переработать систему авторизации. В свою очередь, перенос личного кабинета читателя СФУ на другую платформу влечет за собой немалые трудозатраты в первую очередь из-за большого количества разработанных разделов. Особенность личного кабинета читателя СФУ в том, что за последние несколько лет было разработано достаточно большое

количество разделов и разработки продолжают до сих пор. Использование более новых программных продуктов J-IRBIS и ИРБИС128 теоретически дает возможность решить проблему с авторизацией, однако трудовые и финансовые затраты будут достаточно велики. Помимо стоимости самих программных продуктов, затратной части на перенос и поддержку существующего функционала, существует проблема с нагрузкой на сервер, так как и приложение Web-ИРБИС64 и сетевой сервер ИРБИС должны физически выполняться на одной машине. Ни один из перечисленных продуктов не обладает собственной реализацией интерпретатора сценариев Web-ИРБИС64 и языка форматирования, что не позволяет использовать их отдельно от сетевого сервера ИРБИС. J-IRBIS 1.X использует в фоне тот же самый Web-ИРБИС64 и должен физически располагаться на том же сервере, J-IRBIS 2.0 и ИРБИС128 переключают нагрузку по обработке форматов на сетевой сервер ИРБИС64. Кроме этого, в Web-ИРБИС64 используется поиск по диапазону терминов, который реализован средствами библиотеки `irbis64.dll` и отсутствует в языке запросов CDS ISIS, что усложняет реализацию аналогичного функционала исключительно средствами сетевого сервера ИРБИС.

Вследствие этого возникла идея написать собственное WEB приложение, которое было бы совместимо со сценариями Web-ИРБИС64 и САБ ИРБИС64 и имело возможности для решения проблемы с авторизацией читателя через внешние источники. Начало работе было положено путем создания приложения – прокси между WEB сервером и шлюзом Web-ИРБИС64. Данное приложение было предназначено для взаимодействия с внешними источниками данных для авторизации и последующей подмены ключей авторизации Web-ИРБИС64. Для того, чтобы сделать это возможным, понадобилось узнать алгоритм формирования ключа безопасности Web-ИРБИС64.

Алгоритм ключа безопасности Web-ИРБИС64 был получен путем реверс инженерии. Основная часть алгоритма, ответственная за упаковку ключа была извлечена достаточно быстро, однако долгое время не удавалось понять алгоритм создания временной составляющей ключа. Было совершенно очевидно, что это некая производная от системного времени, в первых числах каждого года, она полностью соответствовала времени от начала года в миллисекундах. Однако, примерно каждые 50 дней, это значение обнулялось. Понадобилось около недели работы с ассемблерным отладчиком, чтобы распознать, задокументировать и воспроизвести все блоки кода, выполняющие операции над временной составляющей ключа. Результат работы оказался весьма неожиданным. Оказалось, в код Web-ИРБИС64 включено некое самодельное ПО для работы с датой и временем, возможно также, что это была часть библиотеки VCL Delphi. Обнуление временной составляющей каждые 50 дней было не результатом работы какого-то алгоритма, а банальным усечением 8 байтового числа двойной точности до 4 байтового целого на 32 битном регистре. Результирующий код на C# представлен одной строкой.

Анализ поведения Web-ИРБИС64 с личным кабинетом читателя СФУ показал, что для работы используется исключительно библиотека `irbis64.dll`. Структура файлов приложения Web-ИРБИС64 хорошо задокументирована авторами. Помимо обычных файлов ИРБИС, Web-ИРБИС64 использует файлы \*.FRM с html разметкой и с включениями программных конструкций на языке форматирования, по структуре похожие на файлы со смешанным кодом PHP и ASP.NET. По сути, вся программная логика WEB приложения написана на языке форматирования ИРБИС.

Путем трассировки использования функций библиотеки `irbis64.dll` был выявлен механизм взаимодействия между WEB средой и языком форматов ИРБИС. Все параметры запроса передаются в сценарии на языке форматирования с помощью полей записи. В Web-ИРБИС64 производится обработка двух видов записи – записи читателя и записей из баз данных электронного каталога. Перед выполнением форматов Web-ИРБИС64, дописывает в каждую из них переданные параметры. Большая часть полей с параметрами описана в конфигурационном файле приложения Web-ИРБИС64, недостающие выявлены с помощью трассировки.

Для создания собственной реализации WEB шлюза потребовалось написать интерпретатор документов со смешанным кодом html и pft (файлов \*.FRM). Первоначально в качестве средства для работы с базой данных и выполнения форматов была задействована библиотека `irbsi64.dll`.

Первый же опыт эксплуатации приложения в среде ASP.NET выявил слабые места такого подхода. В рамках архитектуры .NET библиотеку `irbis64.dll` напрямую использовать нельзя. Библиотека написана на Delphi, экспортируемые функции реализованы по стандартной для Delphi схеме

fastcall с распределением передаваемых в функцию параметров по регистрам процессора. Данный метод передачи параметров невозможно использовать в .NET. Для того, чтобы использовать данную библиотеку, пришлось создать библиотеку – обертку, которая реэкспортирует все функции с помощью stdcall (передача параметров через стек). На этом трудности не закончились. По мере работы на некоторых форматах, которые без проблем обрабатываются самим Web-ИРБИС64 и сетевым сервером ИРБИС обернутая библиотека irbis64.dll начала вызывать разнообразные ошибки. Уходить в бесконечный цикл, вызывать переполнение стека, вызывать аварийное завершение приложения с разнообразными кодами возвратов. Периодические сбои на форматах, которые сами по себе не вызывают ошибок также были частым явлением. Допустим, в ситуации, когда требуется выполнить формат для сортировки набора записей 1000 раз, часто примерно после 700 итерации приложение аварийно завершалось. Выявить какую-либо закономерность сбоев при этом не удавалось.

На фоне проблем с эксплуатацией irbis64.dll возникла идея найти открытую реализацию языка форматирования CDS ISIS (предшественника ИРБИС). Были рассмотрены проект open isis, также программный код утилит по работе с базами CDS ISIS, который выкладывает компания BIREME (разработчик CDS ISIS). Помимо этого, рассматривался проект Алексея Миронова ManagedIrbis. В рамках данного проекта Алексей разрабатывает ПО для взаимодействия с базами ИРБИС на платформе .NET, однако на тот момент рабочей версии языка форматирования в нем не было. В итоге был найден проект J-ISIS, разрабатываемый под эгидой unesco. В рамках данного проекта разработчики создали с нуля на платформе netbeans программное решение, совместимое с CDS ISIS. Приложение написано на java 8, создано кроссплатформенным со встроенной поддержкой utf-8.

После беглого просмотра возможностей J-ISIS были обнаружены хорошо знакомые по работе с ИРБИС файлы форматов, таблицы выбора полей, рабочие листы. В интерфейсе J-ISIS можно выполнять как имеющиеся в базе данных форматы, так и форматы, создаваемые с помощью текстового редактора на лету. По синтаксису язык форматирования J-ISIS очень напоминает язык форматов ИРБИС, так как оба этих языка унаследованы от языка форматирования CDS ISIS. Проект J-ISIS позиционируется как проект с открытыми исходными кодами, однако, в репозиториях на GitHub реализация языка форматов отсутствует. Она оказалась частью базовой библиотеки проекта, которую, по неким причинам, авторы не выложили. Чтобы проверить возможности языка форматов пришлось воссоздать исходные коды базовой библиотеки на java с помощью реверс инженерии. Тесты совместимости языка форматов J-ISIS с языком ИРБИС выявили ряд существенных различий по поведению и некоторые различия в синтаксисе. Тем не менее, реализация языка J-ISIS оказалась наиболее перспективной.

По мере разработки версии языка форматирования ИРБИС на C# с использованием опыта J-ISIS были выявлены многочисленные баги уже в самом языке J-ISIS. Разработчик J-ISIS Jean-Claude Dauphin проявил живейший интерес к информации о найденных проблемах, в достаточно сжатые сроки внес предложенные изменения в код. В релизе J-ISIS 1.2 об этом есть упоминание в разделе **Fixes to the J-ISIS Print Format** <https://github.com/J-ISIS/J-ISIS/releases/download/v1.2/J-ISIS.release.1-2.final.pdf>. Разработчик J-ISIS также заинтересовался функцией по работе с глобальными переменными. Помимо этого я реализовал синтаксис прямого присваивания значений полей, подполей и отдельных фрагментов параллельно для нового языка форматирования на платформе .NET и для J-ISIS. Внедрение данной возможности в J-ISIS обсуждалось в ноябре 2017 года, однако, к сожалению, после этого активность по проекту J-ISIS прекратилась в силу личных причин разработчика.

Данный проект получил название WebIrbisNet. Программный код по взаимодействию с ИРБИС частично основан на переработанной версии проекта Алексея Миронова ManagedClient и отдельных элементах проекта ManagedIrbis. Основа библиотеки unifog и umarcsi взята из проекта ManagedIrbis. Помимо исходных кодов Алексей Миронов оказал неоценимую помощь в прояснении различных аспектов работы ИРБИС, особенностей работы языка форматирования. Впоследствии, часть функций unifog, взятых из проекта ManagedIrbis переписана, часть реализована с нуля по документации и с помощью реверс инженерии irbis64.dll. Также с помощью реверс инженерии выявлено и в основной массе реализовано около двух десятков недокументированных функций

unifor. Были также реализованы точные копии функций ГПНТБ по постобработке расформатированного текста, очистке от ttf конструкций и тд. По опыту эксплуатации данного ПО с базами данных СФУ был разработан механизм универсального декодирования текста, который понимает, как фрагменты с utf-8, так и с кодировкой windows в одном файле.

Для совместимости с форматами ИРБИС в язык форматирования J-ISIS были внесены изменения, введенные разработчиками ИРБИС:

- глобальные переменные
- «висящие» литералы, в том числе после фиктивных команд вывода
- вывод лишнего повторения в повторяющейся группе
- контроль работы повторяющихся групп на основе наличия вывода и бесконечный цикл по этой же причине с отсечкой после 5000 итераций
- остановка повторяющихся групп по команде break
- измененное поведение функции F
- пустые блоки if then else
- синтаксис включения текста внешних форматов реализован как альтернатива стандартному синтаксису запуска на выполнение внешних форматов CDS ISIS. После тестирования внешних форматов в режиме слияния текста, как это делает ИРБИС, было решено от такого подхода отказаться в пользу оригинальной реализации J-ISIS, которая является более производительной.

В целях совместимости поведения форматов воспроизведены баги языка форматирования ИРБИС, являющиеся частью ожидаемого функционала:

- баг в функциях P и A при работе с пустыми подполями
- игнорирование пустых повторений полей, в том числе подавление вывода безусловных литералов и перевода строки
- прерывание работы ТВП в unifor 8 (см в списке недокументированных) после вызова unifor K, unifor +O, unifor 6
- прерывание выполнения формата после деления на 0
- баг со сбросом индекса первого символа подстроки после встречи пустого повторения подполя

К сожалению, логика ИРБИС по обрезанию пустых подполей в режиме данных, отличается от CDS ISIS и никак не отражена в документации, поэтому воспроизвести ее не удалось.

По согласованию с разработчиком J-ISIS, был разработан синтаксис прямого присваивания значений полей / подполей в записи и в глобальных переменных. Была разработана реализация таблиц выбора полей в обоих форматах FST и IFS, реализация глобальных корректировок. Разработана имплементация алгоритма актуализации данных в файлах ИРБИС, в том числе актуализация инвертного файла. В архитектуре J-ISIS язык форматирования отвязан от конкретной реализации поставщика данных. Для работы программы на языке PFT требуется запись данных и несложный интерфейс для взаимодействия с базой. Это позволяет использовать функциональность языка форматирования с различными источниками данных. В рамках проекта WebIrbisNet реализованы поставщики данных как для работы с локальными файлами данных ИРБИС, так и удаленными базами с помощью сетевого клиента. Помимо этого разработан интерфейс для работы с реляционными СУБД и конкретная реализация для PostgreSQL.

Воспроизведен когда-то известный в CDS ISIS механизм использования функций из внешних dll в качестве форматных выходов. В ИРБИСе данный механизм используется для запуска самих функций unifor и umarcsi из самой библиотеки irbis64.dll, а на замену его сделан unifor +8. В рамках данного проекта такая возможность была вновь добавлена с помощью использования .NET плагинов. Реализовано использование плагинов для кодирования / декодирования ключа безопасности Web-ИРБИС64, хранения кэша результатов поисковых запросов, результатов сортировки и фильтрации списка с помощью формата. Функция кэширования результатов поиска встроена также и в unifor +3, который часто используется внутри форматов Web-ИРБИС64. Язык форматирования J-ISIS оказался менее толерантен к ошибкам синтаксиса, которые часто встречаются в форматах (отсутствующий или лишний блок fi, незакрытые скобки в функциях, отсутствие пробелов между

инструкциями). С одной стороны это вызывает некоторые сложности при эксплуатации, с другой, помогает отлаживать недочеты в существующих форматах и ТВП. Разбор больших файлов форматов происходит медленно, поэтому реализована функция хранения разобранных форматов в файловом кэше. Разобраный формат представляет собой список экземпляров классов команд, которые расположены в той же библиотеке (абстрактное синтаксическое дерево – AST), выполнение форматов происходит с JIT компиляцией и в целом работает достаточно быстро. Помимо JIT при работе приложения WEB шлюза активно используется параллельное выполнение форматов на больших списках, это дает свои преимущества. При выполнении запроса к WEB шлюзу с отображением большого количества записей с сортировкой, расформатированием через большой формат, такой как FullWebr.pft преимущества такого подхода становятся очевидны.

Реализован язык запросов CDS ISIS, который в ИРБИС64 реализован не полностью, возможно, по этой причине Web-ИРБИС64 не пользуется сетевым сервером, а работает напрямую с базами с помощью irbis64.dll. Также по опыту эксплуатации WEB шлюза в язык запросов CDS ISIS добавлен полноценный оператор для поиска по диапазону терминов, что позволило реализовать возможность использования Web-ИРБИС64 с удаленными источниками данных, включающими как сетевой сервер ИРБИС, так и реляционные СУБД. Предварительно реализован вариант языка запросов в стиле SQL с использованием выражений PFT.

Данное программное обеспечение реализовано как базовая библиотека для платформы .NET, подобная irbis64.dll. Помимо приложения WEB шлюза, совместимого с Web-ИРБИС64, с использованием данного ПО, разработана версия сетевого сервера ИРБИС, которая может эксплуатироваться с различными хранилищами данных как в операционных системах семейства windows, так и в unix.

## Интеграция с unix

Запуск приложений, написанных на .NET в операционных системах семейства unix возможен двумя способами.

- Запуск с помощью среды mono. Данный проект существует уже более 10 лет. Первоначально спонсировался Novell, сейчас это компания Xamarin, которая недавно была выкуплена Microsoft. Проект mono остается открытым, на его основе Xamarin выпускает одноименную среду для программирования на .NET для мобильных приложений. Приложение mono может выполнять бинарный код, собранный как собственным компилятором, так и компилятором Microsoft из состава .NET SDK. В настоящее время пакет mono включает себя поддержку версий .NET с 1.1 по 4.7. Подходит для разработки как консольных, так и сервисных приложений. Имеет поддержку ASP.NET 4.0 и ASP.NET mvc 4 и 5. Степень реализации в mono технологий, присутствующих в .NET можно увидеть на сайте <http://www.mono-project.com/docs/about-mono/compatibility/>

- Запуск с помощью среды .NET CORE. Сравнительно молодая среда разработки, поддерживаемая Microsoft. Разрабатывается с поддержкой как ОС семейства Unix, так и облачных сервисов Azure. Поддерживаются консольные приложения и ASP.NET 5. Подходит для создания консольных приложений в среде Unix и использования их в качестве сервисов. WEB приложения возможно создавать с помощью ASP.NET 5. Для запуска приложений в .NET CORE необходима сборка целевого приложения компилятором из состава .NET CORE SDK.

Данное программное обеспечение помимо классической среды .NET 4.0 тестируется и в кроссплатформенных средах mono и .NET core, что позволяет эксплуатировать решения, построенные на его базе в операционных системах семейства Unix.

Предварительно проработан вопрос интеграции данной библиотеки со средой PHP. Сделаны модули интеграции в среду Phalanger (среда php 5 для ASP.NET от DevSense), а также расширение для оригинального PHP 7 с использованием библиотек mono для выполнения кода .NET.

Работающий WEB шлюз СФУ, сделанный в рамках проекта WebIrbisNet расположен по адресу [http://catalog.sfu-kras.ru/webirbisnet/cgi-bin/irbis64r\\_14/cgiirbis\\_64.exe?LNG=&C21COM=F&I21DBN=BOOK1&P21DBN=BOOK1](http://catalog.sfu-kras.ru/webirbisnet/cgi-bin/irbis64r_14/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK1&P21DBN=BOOK1)

Приложение WEB шлюза реализует следующие команды Web-ИРБИС64:

- команда вывода HTML страницы F
- команда поиска S
- команда показа терминов словаря T
- команда вывода количества найденных документов G
- команда загрузки бинарного ресурса из записи Z
- функция загрузки локальных файлов Z. Также при реализации исправлен баг Web-ИРБИС64а, позволяющий скачивать любой файл из системы.

### **Интересные изменения в синтаксисе языка форматирования**

J-ISIS разрабатывался на основе наиболее поздней реализации продукта WinISIS – наследника CDS ISIS, вследствие чего в синтаксисе языка форматирования имеются конструкции, которых нет в ИРБИС. Это числовые и строковые переменные, операторы циклов, также появились дополнительные функции, команды обновления значений полей в стиле CISIS. Вот несколько примеров форматов для иллюстрации.

#### **Вывод всего списка результатов поиска с явным указанием базы данных**

```
ref->ИМЯ_БД(lr->ИМЯ_БД(('ТЕРМИН')),v200)
```

#### **Вывод части списка с указанием диапазона**

```
ref->ИМЯ_БД(lr->ИМЯ_БД(('ТЕРМИН'), 10, 20),v200)
```

#### **Вывод списка результатов поиска в цикле**

```
e1:=1,e2:=(npst('ТЕРМИН')), while e1<=e2 (f(e1,1,0),' -> ',ref(lr(('ТЕРМИН'),e1,e1),v200) / e1:=(e1+1))
```

#### **То же самое, с указанием имени базы данных**

```
e1:=1,e2:=(npst->ИМЯ_БД('ТЕРМИН')), while e1<=e2 (f(e1,1,0),' -> ',ref->ИМЯ_БД (lr->ИМЯ_БД  
(('ТЕРМИН'),e1,e1),v200) / e1:=(e1+1))
```

В данном примере используется усовершенствованный синтаксис, который был добавлен уже в рамках проекта WebIrbisNet. В J-ISIS сборки ноября 2017 это работать не будет. В частности, на момент ноября 2017 года в J-ISIS не поддерживается использование переменных в параметрах, указывающих на диапазон записей функции LR, в функции NPST не предусмотрено явное указание имени базы данных.

#### **Синтаксис работы с повторениями подполей в том же стиле, что и с повторениями полей**

```
g100[2..3]^a[2..3]*2.2
```

#### **Прямое присваивание значений полей, подполей, в записи и в глобальных переменных**

Новый синтаксис присваивания значений полей / подполей, реализованный мной для проекта J-ISIS . Сделан с поддержкой повторяющихся групп.

Запись, использованная в примере.

```
25#^aB014809^b123
```

```
25#^aB014806
```

### Прямое копирование указанных повторений полей

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

Вывод:  
B014809  
B014806

### Копирование результатов расформатирования

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

Вывод:  
B014809  
B014806

### Копирование поля целиком со всеми повторениями

v31:=(v25),(v31 /)

Вывод:  
^aB014809^b123  
^aB014806

### Копирование всех повторений поля с сохранением старого значения

v31:=('1 occ'),(v31[2]:=(v25)),(v31 /)

Вывод:  
1 occ  
^aB014809^b123  
^aB014806

### Копирование всех повторений поля с сохранением старого значения в повторяющейся группе

(v31:=(v25)),v31[3]:=('3 occ'),e1:=nocc(v25),(if occ <= e1 then v31[4]:=(v25) fi), (v31 /)

Вывод:  
^aB014809^b123  
^aB014806  
3 occ  
^aB014809^b123  
^aB014806

### Замена фрагментов подполей

g100:=("),g100[3..4]^a[2..3]\*2.5:=('12345'/6789'),g100[2..3]^a[2..3]\*2.2:=('12345'/6789'),(g100 /)

Вывод:  
^a^a12345^a  
^a^a1267895^a  
^a^a6789^a

## Вызов плагина из языка форматов

Пример кода на языке C#.

```
//регистрируем плагин
FormatExit.Register("date",
(ctx, s) =>
{
//форматируем текущую дату с переданной строкой формата
var fmt = String.Format(s, DateTime.Now);
ctx.Print(fmt);
});

//открываем базу данных
var db = newIrbis64ManagedDatabase("IBIS");
//создаем пустую запись в памяти
db.NewRecord();
//выполняем формат
var format = db.Format("v100[1] := (&date('текущая дата ... {0:yyyy.MM.dd}')), v100[2] := (&date('время ...
{0:HH.mm.ss.fff}')), (v100 /)");
Console.WriteLine(format);
```

Вывод:

текущая дата ... 2018.04.04

время ... 13.35.50.503

## Изменения в языке запросов

В сетевом сервере ИРБИС и в Web-ИРБИС64 используется реализация языка запросов CDS ISIS. В Web-ИРБИС64 существует операция поиска по диапазону года издания, например с 2004 по настоящее время. Данную операцию невозможно реализовать с помощью канонического языка запросов CDS ISIS. В кабинете читателя СФУ расшифровка по данному запросу выводится в виде ГОД ИЗДАНИЯ NNNN [...] NNNN. Реализация данного запроса добавлена в синтаксис языка запросов. Пример запроса, который может обрабатываться напрямую

```
("K=ПИВО")*(K=ВОДЫ)*"G=2004"[...]G="*"M=СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ$"
```

Реализация данного вида запроса возможна с помощью типового API по работе с инвертным файлом, вследствие того, что термины упорядочены. Введено ограничение, чтобы оба термина имели общий префикс, хотя это не является принципиально важным. Для обработки такого запроса выполняется поиск по первому термину, это позволяет найти позицию первого ключа в L01. Далее идет простой цикл по ключам, пока ключ не выйдет за границу второго термина, либо общего префикса, как в указанном примере. Реализация данного алгоритма также сделана с помощью сетевого клиента, однако в этом случае процесс манипуляции с ключами выполняется заметно дольше.

Помимо оператора поиска по диапазону реализован четвертый оператор контекстного И языка запросов CDS ISIS, который отсутствует в ИРБИС.

Далее перевод документации win isis.

Знак доллара \$, обрамленный пробелами – четвертый оператор контекстного И. Соединение двух терминов таким оператором контекстного И обозначает требование поиска записей, в которых оба термина присутствуют в одном и том же повторении поля друг за другом (или точнее – когда у терминов совпадают вторые и третьи части ссылок, а четвертые части ссылок отличаются на количество знаков \$). В win isis учитывается также количество знаков \$. Оператор \$ ведет себя также как (F), однако имеет дополнительное ограничение, что термины должны иметь строго n слов между ними, где n – количество знаков доллара минус один. Примеры:



- water \$ soil слова следуют друг за другом
- water \$ \$ soil одно слово между water и soil
- water \$ \$ \$ soil два слова между water и soil

## Неописанные функции unifor

Неописанные коды unifor, выявленные с помощью реверс инженерии библиотеки irbis64.dll.

- **unifor** , заставить работать не удалось
- **unifor** . очистка двойных точек в unifor со вложенным расформатированием, заставить работать не удалось
- **unifor A** сама функция описана, однако есть неописанные опции, I ведет себя также как V, D то же, что V, но номер повторения с конца
  - **unifor P** все опции I,V,G,D игнорируются, ведут себя как V
  - **unifor H** извлекает текст, заключенный в угловые скобки <> вместе со скобками
  - **unifor 3M** дата из числа в днях от базовой даты 30.12.1899
  - **unifor +3A** вывод полей текущей записи в клиентском представлении, только поля без заголовка записи
  - **unifor +3F** поиск и расформатирование записей. Аналогична unifor +3S, только поисковый запрос делится на 2 части тем же разделителем. Вторая часть – уточняющий запрос с учетом морфологии. Также как unifor +3S сбрасывает флаги постредактуры в контексте
  - **unifor +3J** возвращает количество ссылок на термин. Термин ищется без обязательного точного совпадения, но требуется, чтобы указанный текст присутствовал в термине. Параметры DBNAME,TERM если база данных не указана, берется текущая, если указать имя другой базы – не работает. Точнее, ИРБИС ищет в DATAI файл .PAR указанной базы, если находит, то ничего не делает. Можно указать другую базу по относительному пути от DATAI, например в виде DBNAME/DBNAME, тогда ИРБИС будет использовать инвертный файл указанной базы. Аналогичен unifor J, в отличие от последнего поиск идет не полному совпадению, а по вхождению искомого термина в найденный отличается способ передачи параметра базы данных
  - **unifor +3T** делит строку на 2 фрагмента по символу запятой, разбирает как double, делит, возвращает целую часть результата
  - **unifor +3H** реализация функции PHP htmlspecialchars, заменяет
    - & на &quot; (здесь ошибка надо на &amp;)
    - " на &quot;
    - < на &lt;
    - > на &gt;
    - одинарные кавычки не кодирует
  - **unifor +9B** неописанный клон функции +98
  - **unifor +9E** преобразование представленного в параметре числа как размера файлов чело-веко-читаемую строку с b, Kb, Mb
  - **unifor +9H** параметры аналогично +9S, выполняет конкатенацию строки А и Б в случае наличия Б. Если строки Б нет, выводит пустую строку
  - **unifor +9N** поиск следующего термина
  - **unifor +9P** поиск предыдущего термина
  - в обоих случаях +9N,+9P параметры одинаковые, такие же, как в +3J
  - **unifor +9T** выводит список чисел от А до В. параметры А/В или А. Числа выравниваются нулями по длине А, отрицательные числа также выравниваются нулями перед знаком минус
  - **unifor +9Z** прочитать поле из справочника текущей записи с номером, который возвращает Irbisfieldn, параметр – число от 1 до Irbisfieldn. Параметр парсится как число с 1 по умолчанию
  - **unifor +O** неописанный клон функции K
  - **unifor +T** заставить работать не удалось, судя по коду – пустышка

- **unifor +H** перебирает строку по байтам. Берет 3 байта, 4-й отбрасывает, в начало помещает количество таких групп (по 4 байта)

- **unifor +C** ищет в начале или в конце строки цифры, парсит как int32, прибавляет 1 и заменяет

- **unifor +K** поиск кода элемента в меню по значению, список должен быть упорядочен, возможно, предназначен для авторского знака

- **unifor +Z** преобразование DOS в ANSI и наоборот. Если первый символ 0 вызывает CharToOemA, если первый символ 1 вызывает OemToCharA\_0

- **unifor +B** суммирует байты строки параметров, видимо аналог контрольной суммы

- **unifor +M** очистка пустых повторений 991 поля в записи на полке с помощью функции Irbisfldrep

- **unifor +X** параметры A#B ищет термин A, идет циклом по терминам пока текущий термин не станет дальше по сравнению с ним, контекст не закрывает. Дальше вызывает ту же функцию, что **unifor +W**

- **unifor +W** похоже на unifor +C. Параметры A#B берет параметр B, ищет группы цифр в любом месте текста, если группа одна, парсит как int32, прибавляет 1. Если число не умещается в int32, не делает ничего. Если переполнение int32, выводит отрицательное значение. Отрицательные числа не обрабатывает.

- **unifor +L** test.mnu|123. Ищет в базе (только в базе, в депозит не смотрит) файл test.mnu. Читает как utf8, берутся строки кроме первой и последней. Ищет в файле строки равные подстроке параметра побуквенно. В указанном примере ищет строки 1, 12, 123. В вывод попадают максимум 6 строк. Строки идут в порядке, как идут в фильтре буквы, шестой идет последняя найденная. Найденные строки идут с подполями A B C D F G

- **unifor +V** выводит подстроку указанной длины. Параметры ДЛИНА#СТРОКА

- **unifor +P** обертка для функции Irbisposting, параметры POSTING, ТЕРМИН, работают варианты постинга 1-3, 4 почему-то не поддерживается. В функции 0 подставляется 1 в Irbisposting, 1 – 2, 2 – 3

- **unifor ++A** ищет 4 или более буквы подряд (кириллица, латиница) в верхнем регистре, возвращает подстроку, начиная с них, символы других алфавитов ломает

- **unifor 8** формат 8<dbn>,<@mfntermin/>,<fst>,<tag>,<teq>

Передаются пять параметров, разделенные запятой:

- Первый – имя БД;
- Второй – или непосредственно MFN с предшествующим
- символом @ или термин, ссылающийся на документ
- (термин – заключается в ограничительные символы);
- Третий – имя файла ТВП в формате FST (IFS не поддерживается)
- Четвертый – тег из FST
- Пятый метод индексирования

Читает FST, ищет строки с указанным тегом и методом,

расформатирует найденную запись

прерывает выполнение, если в формате текущей строки выполнялись

**unifor K, unifor +O, unifor 6**

В 2017 версии добавлено

- **unifor** : параметры шаблон#строка.

Сравнивает первые 4 группы символов, разделенные точкой.

В шаблоне подстрока \* означает любую последовательность символов.

Похоже на анализ ipv4, но без проверки цифр.

Пример 123.aa.\*.1#123.aa.10.1

Группы символов 5 и далее отбрасывает.

В случае соответствия шаблону выводит 1 иначе 0

- **unifor ++B** установка времени сеанса читателя

<http://irbis.gpntb.ru/read.php?3,94172,96120#msg-96120>

имеет смысл только при наличии полной собственной реализации

- **unifor ++C** глобальный счетчик в базе COUNT <http://irbis.gpntb.ru/read.php?3,94172>

## Интеграция с PHP

Взаимодействие с языком PHP возможно 3 способами:

- Запуск языка PHP средой Phalanger. Это компилятор языка PHP 5 с открытыми исходными кодами для ASP.NET. Компилирует скрипты PHP в модули .NET и использует их со своими рантайм библиотеками. Умеет подключать оригинальные расширения PHP. Работает достаточно быстро, однако не всегда имеются реализованные функции, иногда требуется доработка. Модуль ИРБИС для Phalanger реализован.
- Запуск модуля ИРБИС в качестве расширения PHP с помощью механизма встраивания среды топо. Реализовано расширение, поставляющее функцию для запуска обработчика Web-ИРБИС64 в виде функции PHP. Работает в рамках самостоятельного процесса FastCGI, в составе модуля Apache среда топо не запускается.
- Данный подход пока существует в рамках идеи. Успешный опыт реализации базы данных на C#, которая локально исполняет форматы и работает с удаленным сервером с помощью сетевого клиента, показывает возможность повторить то же самое на PHP. Сетевой клиент ИРБИС для языка PHP существует и эксплуатируется в рамках ИРБИС128. Потребуется реализовать на PHP классы AST языка форматов. Транслятор форматов в дерево AST может быть сделан как в виде внешней утилиты на основе данного проекта, так и в виде расширения PHP на языке C/C++. Разобранное дерево AST формата можно хранить в виде PHP файлов и использовать по мере необходимости. С помощью сетевого клиента будут осуществляться операции по работе с записями, поиск, загрузка файлов с сервера ИРБИС.

## В качестве эпилога

Платформа ИРБИС аккумулирует передовой опыт в разработке библиотечных решений, является простой и надежной в эксплуатации, а также недорогой системой. Не требует участия высокооплачиваемых специалистов. К недостаткам можно отнести недостаточную развитость средств разработки. Данное исследование призвано показать возможность быстрого создания расширяемой платформы, которая поможет исправить данную ситуацию и вынести тему взаимодействия с ИРБИС в мир популярных средств разработки, таких как PHP, .NET, JAVA и т.д. В ближайших планах видится создание кроссплатформенных инструментов для работы с форматами ИРБИС и фреймами Web-ИРБИС64 для языка PHP, в рамках расширений языка PHP для запуска модулей .NET, реализации транслятора AST или в виде отдельного сервера форматов, пригодных для использования в других проектах, таких как ИРБИС128 и J-IRBIS. Также интересуют тема реализации альтернативных хранилищ данных в различных СУБД и проработка полной реализации серверной части, совместимой с другими компонентами ИРБИС для ОС семейства Unix. Помимо этого, хотелось бы наладить взаимодействие с разработчиками ИРБИС для обретения этими разработками официального статуса и проработки взаимодействия.