

Основы макроязыка EASI (PCI Geomatica) Часть 1.

Управление данными

[Обсудить в форуме](#) Комментариев — 0

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/easi1.html>

Первая часть серии статей о языке EASI, основные команды языка по работе с растровыми данными.

Содержание

- [1 Введение](#)
- [2 Основы работы с языком EASI](#)
- [3 Основы создания скриптов](#)
- [4 Особенности работы с файлами PCIDSK](#)
- [5 Команды для создания файлов](#)
 - [5.1 CIM – создание файла PCIDSK](#)
 - [5.2 CIMPRO2 – создание файла и назначение ему проекции](#)
- [6 Команды импорта/экспорта растровых данных](#)
 - [6.1 FIMPORT – импорт файла](#)
 - [6.2 FEXPORT – экспорт файла](#)
 - [6.3 LINK – «импорт» файла в формат PCIDSK без переноса растровых данных](#)
 - [6.4 Другие полезные команды](#)
- [7 Команды удаления/добавления каналов и сегментов](#)
 - [7.1 PCIADD2 – добавление нового канала](#)
 - [7.2 PCIDEL2 – удаление канала](#)
 - [7.3 DAS – Удаление сегмента](#)
- [8 Команды обмена данными между файлами PCIDSK](#)
 - [8.1 IIA – Перенос сегмента](#)
 - [8.2 IIB – Перенос растровых масок](#)
 - [8.3 III – Перенос канала изображения](#)
 - [8.4 IIIAVG – Перенос канала изображения с усреднением значений](#)
 - [8.5 MOSAIC – Создание бесшовной мозаики изображений](#)
- [9 Команды чтения информации о структуре файлов PCIDSK](#)
 - [9.1 CDL – Вывод на экран \(или в файл\) списка каналов изображения](#)
 - [9.2 ASL - Вывод на экран \(или в файл\) списка сегментов файла PCIDSK](#)
 - [9.3 IHR – Чтение и запись в набор переменных информации о структуре файла и привязке](#)
- [10 Ссылки по теме](#)

Введение

Критерии практической ценности различных систем обработки дистанционных данных постоянно меняются, особенно в России. Еще несколько лет назад основным критерием выбора программного обеспечения был набор функциональных возможностей и алгоритмов обработки данных. Однако в последние годы две тенденции: усиление контроля за лицензированием ПО и лавинообразный рост количества доступных данных требуют изменить сложившиеся представления. Возможно, специалистам, ранее пользовавшимся дорогими (но не лицензированными) системами «высшего уровня», типа ERDAS Imagine, следует подумать о приобретении легальных копий более дешевых систем; а вместо ориентации на индивидуальную обработку снимков, подумать об автоматизации процесса обработки.

Одной из достаточно дешевых полнофункциональных систем обработки изображения является канадская PCI

Geomatica (<http://www.pcigeomatics.com>). Пользователям продуктов ESRI и ERDAS интерфейс программы может не понравиться своей необычностью и небольшим набором встроенных функций. Большинство операций обработки данных осуществляются с использованием конструктора моделей (Modeler) и встроенного макроязыка EASI. Однако именно опора на использование языка командной строки в Geomatica делает автоматизацию процессов чрезвычайно простой. Дополнительный значительный плюс языка EASI – возможность запуска интерпретатора из командной строки с именем файла скрипта в качестве параметра (easi r myscript.eas). Это позволяет с легкостью создавать длинные цепочки операций, в которых скрипты EASI могут создаваться и запускаться из других оболочек ([например, PERL](#)). Geomatica поставляется как для ОС Windows, так и для Linux, что позволяет использовать ее на мощных серверах обработки данных.

В данной статье рассмотрены особенности работы с языком EASI и приведено описание основных команд по созданию и управлению данными. В следующих статьях мы остановимся на [анализе данных](#), [привязке и работе с проекциями](#) и возможностях [автоматизации](#).

Основы работы с языком EASI

Запуск и правила использования команд в языке EASI чрезвычайно простые. Интерпретатор запускается как из оболочки (панель задач Geomatica) так и из командной строки. Мы предлагаем по возможности использовать второй путь, не загромождая память компьютера интерфейсом оболочки системы. Итак, из рабочей директории набираем:

```
easi
```

Первое сообщение при запуске EASI из новой директории «отсутствует файл параметров!» Чтобы скопировать файл параметров в рабочую директорию, набираем

```
r copprm
```

В рабочей директории будет создан файл параметров PRM.PRM.

Собственно, теперь можно выполнять команды. Для выполнения команд можно использовать три ключа, добавляемых перед именем команды:

- h (help) – вызов справки о команде
- s (status) – вызов списка параметров команды
- r (run) – запуск команды на выполнение

Параметры команд вводятся до их запуска. Например:

- file="new – имя файла «new.pix» (расширение собственных файлов PCI можно опускать всегда, когда нет файлов других типов но с тем же именем в той же директории)
- dbsz=10,10 – размер файла 10x10 пикселей
- dbnc=1,0,0,0 – 1 8-битный канал
- dblayout="file – разделять каналы по отдельным файлам
- r cim – теперь запустим команду. Она будет использовать параметры, введенные заранее.

Следует обратить внимание, что значения параметров сохраняются в созданном файле параметров PRM.PRM. Это может сильно сэкономить время для ввода параметров к последующим командам. Однако всегда будьте внимательны и проверяйте значения параметров перед запуском команд. Для проверки набираем:

```
s cim
```

На экран будут выведены параметры команды и их значения.

Для выхода из интерпретатора наберите exit.

Основы создания скриптов

Скрипты являются последовательностью параметров и команд – в точности так же, как если бы они вводились

в командной строке. В начале скрипта полезно вставлять `r corrpm` на случай, если вы скрипт запускаете в новой директории. Скрипт EASI имеет расширение `.eas`. Например `myscript.eas`.

Пример содержание скрипта:

```
file="new
dbsz=10,10
dbnc=1,0,0,0
dblayout="file
r cim
```

Никаких специальных команд начала и завершения скрипта не используется.

Запуск скрипта из среды интерпретатора выглядит как запуск команды:

```
r "myscript.eas
```

В среде Windows можно опустить кавычки и расширение:

```
r myscript
```

Запуск из командной строки (или из среды другой программы) выглядит так:

```
easi r myscript
```

Подробнее на использовании скриптов и специальных функций языка в скриптах мы остановимся в [отдельной статье](#).

Особенности работы с файлами PCIDSK

Работать с данными в PCI Geomatica удобнее, используя собственный стандарт данных PCIDSK. Файлы PCIDSK включают в себя собственно растровые изображения (каналы, raster layers, image channels) и дополнительные данные в виде сегментов (segments).

Каналы характеризуются радиометрическим разрешением (8 bit, 16 bit unsigned, 16 bit signed, 32 bit real). В одном файле могут содержаться каналы с разным радиометрическим разрешением (в отличие от ERDAS Imagine). В файле данных каналы могут храниться в основном файле с расширением `.rix` или во внешних файлах, имеющих то же имя и расширение `.001`, `.002` и т.д. Если каналы хранятся в одном файле, то возможны два варианта хранения каналов: разбивка по пикселям (как в формате BIP) или разбивка по каналам (BSQ). Следует заметить, что при хранении каналов в одном файле часто возникают сбои при попытках создания или удаления каналов. Мы советуем всегда использовать внешние файлы для хранения каналов.

Сегменты хранятся в основном файле с расширением `.rix`. Сегменты отличаются порядковыми номерами (нумерация сегментов непрерывная) и трехбуквенными кодами, обозначающими тип сегмента (трехбуквенные коды эквивалентны численным кодам, которые используют некоторые команды). И каналы, и сегменты имеют текстовые метки имени и описания.

Основные типы сегментов:

- **Georeferencing Type 150:GEO** — этот тип сегментов хранит информацию о привязке и проекции изображения. Сегмент №1 всегда является сегментом привязки. Однако могут быть созданы и дополнительные сегменты, которые могут включать информацию о проекциях, в которые будет трансформировано изображение
- **Bitmaps Type 101:BIT** — растровые маски имеют такое же пространственное разрешение и привязку, что и каналы изображения, однако имеют значения только 1 и 0 (аналогичны 1-битному изображению в ERDAS Imagine). Это один из наиболее часто используемых типов сегментов
- **Vectors Type 116:VEC** — сегменты, содержащие векторные данные. Подразумевается, что проекция векторных сегментов совпадает с проекцией растровых каналов, хотя векторные сегменты могут иметь собственную информацию о проекции, которая может отличаться от основной проекции базы данных

Дополнительные типы сегментов:

- Signatures Type 121:SIG — спектральные сигнатуры классов для классификации
- Look up Table Type 170:LUT — таблицы радиометрической коррекции
- Pseudo-Colour Table Type 171:PCT — таблицы кодирования цветов
- Ground Control Points Type 214:GCP — наборы точек привязки
- Text Type 140:TEX — текстовые примечания

Файл PCIDSK с каналами, разделенными по отдельным файлам, выглядит следующим образом:

```
myfile.pix  
myfile.001  
myfile.002  
myfile.003
```

Для обработки все файлы с одним именем должны храниться в одной директории. Файлы каналов, «оторванные» от основного файла, могут быть импортированы в PCI, если известна структура данных и привязка. Основной файл с расширением .pix может быть использован для операций над сегментами даже в случае отсутствия файлов каналов.

Команды для создания файлов

CIM – создание файла PCIDSK

Использование:

- file= (имя файла)
- dbsz= (размер файла в пикселях (x,y))
- dbnc= (число каналов каждого типов 8U,16S,16U,32R)
- dblayout= (тип хранения каналов в файле pixel/band/file)

```
run cim
```

Пример:

- file="new (расширение можно опустить)
- dbsz=10,10 (10x10 пикселей)
- dbnc=1,0,3,1 (создать 1 8-битный. 3 16-битных и 1 32-битный канал. Каналы будут созданы именно в такой последовательности)
- dblayout="file (разбивка по отдельным файлам)

```
r cim (выполнить команду)
```

CIMPRO2 – создание файла и назначение ему проекции

Использование:

- file= (имя файла)
- dbnc= (число каналов - см. выше)
- dblayout= (тип хранения каналов - см. выше)
- project= (имя проекции)
- zone= (зона проекции UTM)
- ellips= (эллипсоид или датум)
- llbound= (тип координат углов – географические (Y) или метрические (N))
- ulx= (координаты углов)
- uly=
- lrx=
- lry=

- bxpksz=" (размер пикселя по осям. Число пикселей будет определено автоматически)
- bypxsz="
- extrainf= (дополнительная информация о проекции. Команда включает множество дополнительных параметров для разных проекций. Все эти параметры могут быть объявлены «по умолчанию»)

```
r cimpro2
```

Команды импорта/экспорта растровых данных

FIMPORT – импорт файла

Если импортируемый файл растровый, будет создан канал в новом файле. Если импортируется векторный формат – будет создан файл с векторным сегментом. Информация о привязке, если может быть прочитана, будет внесена в сегмент №1.

- fili=" входной файл (с расширением!)
- filo=" (новый файл PCIDSK)
- dbiw= (опционно: может быть задано окно для автоматической обрезки изображения. Окно задается в пикселях изображения. Для векторных форматов не используется)
- dblayout= (тип хранения каналов - см. выше)

```
run fimport
```

FEXPORT – экспорт файла

Возможен экспорт как векторных, так и растровых изображений, а так же специальных сегментов (например, Look up Table)

- fili=" (входной файл)
- filo=" (новый файл, расширение необходимо указать)
- dbiw= (опционно: может быть задано окно для автоматической обрезки изображения)
- dbic= (номера каналов для экспорта. Номера задаются списком: 1,2,3 или интервалом 1,-3)
- dbib= (номера сегментов растровых масок)
- dbvs= (номера векторных сегментов)
- ftype=" (тип файла. например tif, img, shp)

```
run fexportt
```

При экспорте в растровые форматы нельзя указывать векторные сегменты и наоборот.

LINK – «импорт» файла в формат PCIDSK без переноса растровых данных

Эта команда позволяет Вам создать файл .rix с внешним каналом данных, который представляет собой исходный растровый файл. Таким образом могут быть импортированы и файлы с информацией о привязке (например, geotif или ERDAS img).

- fili=" (входной файл)
- filo=" (имя нового файла rix, к которому будет «привязан» входной файл)
- dbiw= (опционно: может быть задано окно для автоматической обрезки изображения)

```
run link
```

Полезное замечание: все операции над «привязанным» файлом будут выполняться аналогично обычному каналу файла PCIDSK. При этом формат файла, имя с расширением и встроенная информация о проекции будет сохранены. Таким образом, по окончании операций, файл можно использовать как обычный geotif или ERDAS img.

Другие полезные команды

NUMWRIT – выгрузка растровых данных в виде текстового файла. NUMREAD – чтение текстового файла в растровое изображение. IMAGERD – импорт бинарных файлов.

Описание этих команд можно найти в справочной системе EASI.

Команды удаления/добавления каналов и сегментов

PCIADD2 – добавление нового канала

- file= (имя файла)
- datatype= (радиометрическое разрешение. Возможные значения: 8u,16s,16u,32r)
- addmode= (create – будет создан пустой канал, exist – канал будет добавлен из другого файла)
- ifile= (Имя файла, который будет добавлен в качестве канала с опцией exist. Например, если существуют два файла с одинаковыми размерами и привязкой, то каналы одного из них (файлы с расширением .001, .002 и т.д.) могут быть привязаны в качестве дополнительных каналов к другому файлу PCIDSK. При этом копирование производиться не будет, что позволит сэкономить место на диске)

```
r pciadd2
```

Полезное замечание: в качестве существующего канала могут быть использованы файлы в формате geotif, img и другие. Совместное использование с командой LINK (см. выше) позволяет Вам собирать «временные» базы PCIDSK из файлов в других форматах для их совместной обработки. По окончании анализа сам файл *.pix может быть удален. Пример формирования базы данных из двух изображений в формате geotif (проекция и размер изображений должны быть одинаковыми!):

```
file="file1.tif  
filo="temp.pix  
r link  
file=filo  
datatype="16u  
addmode="exist  
ifile="file2.tif  
r pciadd2
```

PCIDEL2 – удаление канала

- file= (имя файла)
- delmode= (yes — канал будет физически стерт с диска, no – канал (в виде отдельного файла) сохранится, будет удалена только привязка этого канала к файлу PCIDSK. Эту опцию следует всегда использовать, если Вы использовали каналы другого снимка).
- dbic= (список каналов)

```
r pcidel2
```

Специальной команды создания сегментов типа растровых масок или векторов не существует. Однако часто бывает нужно создать пустую растровую маску для последующего использования в ходе обработки. Обходной путь здесь – использование команды пороговых значений (THR) со значениями, заранее превосходящими амплитуду значений в пределах канала. Например:

- file= (имя файла)
- dbic= (номер пустого канала или канала с известной амплитудой значений)
- dbob= (оставим пустым – этот сегмент будет создан)
- tval= (минимальное и максимальное пороговые значения, заранее больше известной амплитуды данных. Например, если канал, указанный в dbib, пустой, достаточно указать 1,1)
- dbsn= (имя нового сегмента – обязательное поле)

```
r thr
```

Подробнее команда THR будет рассмотрена в [следующей статье](#).

DAS – Удаление сегмента

Номер сегмента останется в базе данных, однако сегмент будет помечен «Deleted», и его номер будет свободным – то есть достанется следующему созданному сегменту.

- fili=" (имя файла)
- dbsl= (номера сегментов)

r das

Команды обмена данными между файлами PCIDSK

IIA – Перенос сегмента

Обычно используется для неграфических сегментов – сегментов привязки, таблиц радиометрической коррекции и пр.

- fili=" (имя исходного файла)
- filo=" (имя файла, куда будет перенесен сегмент)
- dbsl= (номера сегментов)
- dbos= (номера сегментов, на место которых будут записаны перенесенные сегменты. Если поле оставить пустым, будут созданы новые сегменты)

r iia

IIB – Перенос растровых масок

- fili=" (имя исходного файла)
- filo=" (имя файла, куда будут перенесены маски)
- dbib= (номера исходных сегментов)
- dbob= (номера сегментов, на место которых будут записаны перенесенные сегменты. Если поле оставить пустым, будут созданы новые сегменты)

r iib

При переносе растровых данных необходимо, чтобы оба файла имели одинаковый размер (в пикселях). Желательно (но не обязательно) чтобы привязка была одинаковой. Если привязка и размер пикселей совпадают, но линейный размер различается, можно использовать дополнительные параметры:

- dbiw= (размер исходного окна. Размер задается как Xoffset,Yoffset,Xsize,Ysize. Оффсет отсчитывается от верхнего левого пикселя изображения. Все размеры в пикселях)
- dbow= (размер конечного окна. Задается также, как и размер исходного окна) Данный параметр удобен, когда необходимо собрать мозаику изображений без перекрытия.

III – Перенос канала изображения

- fili=" (имя исходного файла)
- filo=" (имя файла, куда будут перенесены каналы)
- dbic= (список исходных каналов)
- dboc= (список каналов в выходном файле. Каналы должны быть уже созданы, параметр обязательно должен иметь такое же число каналов, как число исходных каналов)

r iii

Так же, как и при переносе растровых масок, изображения должны иметь одинаковый размер в пикселях. Возможно использование параметров dbiw и dbow (см. выше).

IIIAVG – Перенос канала изображения с усреднением значений

Усреднение значений выполняется в соответствии с различием в размере пикселей исходного и выходного растров

Параметры команды аналогичны III:

- fili=" (имя исходного файла)
- filo=" (имя файла, куда будут перенесены каналы)
- dbic= (список исходных каналов)
- dboc= (список каналов в выходном файле)

r iiiavg

Для получения корректного результата, растровые сетки должны быть вложенными, т.е. в пределах пикселя выходного файла должно размещаться целое число пикселей исходного файла. Вложенные растровые сетки должны иметь одинаковый размер. Возможно использование параметров dbiw и dbow (см. выше).

Полезное замечание: данная команда удобна для расчета процентного содержания одного класса в пределах пикселей растра с меньшим пространственным разрешением. Для расчета % целевого класса, задайте всем пикселям данного класса значение 100, а прочим пикселям значение 0 в исходном изображении. После выполнения команды IIIAVG выходное изображение будет содержать % класса в пределах каждого пикселя. Команда выполняет округление значения до ближайшего целого в случае использования 8-битного канала. Если Вы хотите сохранить точные значения, канал выходного файла должен быть целочисленным (32r).

MOSAIC – Создание бесшовной мозаики изображений

- fili=" (имя исходного файла)
- dbic= (список исходных каналов)
- filo=" (имя файла мозаики)
- dboc= (список каналов в выходном файле)

```
r mosaic
```

Для уменьшения эффекта перекрытия снимков используются дополнительные параметры:

- dbvs — номер векторного сегмента, где хранится информация о линии разреза растров
- dblut – номер сегмента с таблицей радиометрической коррекции для изменения гистограммы исходного файла перед операцией «склеивания»
- blend – размер зоны перекрытия изображений, в пределах которой значения пикселей будут усреднены
- backval – значение, используемое для обозначения фона мозаики

Полезное замечание: удобная недокументированная возможность команды – перепроецирование данных из проекции исходного растра в проекцию выходного растра. При пересчете значений всегда используется алгоритм «ближайшего соседа» (nearest neighbor). Команда так же может быть использована для «обрезки» растров по размерам, заданным размерами выходного изображения.

Команды чтения информации о структуре фалов PCIDSK

CDL – Вывод на экран (или в файл) списка каналов изображения

Список каналов выводится с кратким описанием (тип данных, имя канала и др.)

- file=" (имя файла)
- report=" (по умолчанию = "term (вывод на экран). Вместо этого может быть задано имя файла: report="report.txt)

```
r cdl
```

ASL - Вывод на экран (или в файл) списка сегментов файла PCIDSK

Список сегментов выводится с кратким описанием (тип данных, имя сегмента и др.)

- file=" (имя файла)
- report=" (term или имя файла)

```
r asl
```

IHR – Чтение и запись в набор переменных информации о структуре файла и привязке

- file=" (имя файла)

```
r ihr
```

Список создаваемых в памяти переменных:

- IMSTAT(1) = размер в пикселях, X
- IMSTAT(2) = размер в пикселях, Y
- IMSTAT(3) = число каналов
- IMSTAT(4) = число каналов 8-bit

- IMSTAT(5) = число каналов 16-bit signed
- IMSTAT(6) = число каналов 16-bit unsigned
- IMSTAT(7) = число целочисленных каналов 32-bit
- IMSTAT(8) = число сегментов
- IMSTAT(9) = размер пикселя, X (м)
- IMSTAT(10) = размер пикселя, Y (м)
- IMSTAT(11) = единицы растра (0=PIXEL,1=UTM,2=METRE,3=LONG/LAT, 4=other,-1=error)
- IMSTAT(12) = Верхний левый угол, X (единицы растра)
- IMSTAT(13) = Верхний левый угол, Y (единицы растра)
- IMSTAT(14) = Нижний правый угол, X (единицы растра)
- IMSTAT(15) = Нижний правый угол, Y (единицы растра)

Полезное замечание: команда IHR удобна, если Вы хотите создать «чистую» копию растра с сохранением размеров и привязки. Вот как будет выглядеть последовательность команд:

```
file="old (имеющийся растр)

r ihr

file="new (создаваемый растр)
dbsz=imstat(1),imstat(2) (используем переменные из памяти)
dbnc= 10,0,0,0 (число каналов по типам)
dblayout="file
r cim (создаем файл)

fili="old
filo="new
dbsl=1
dbos=1
r iia
```

Последняя команда – копирование сегмента привязки. Поскольку создаваемый файл имеет такой же размер, что и исходный, сегменты привязки у них будут идентичными.

Ссылки по теме

- [Основы макроязыка EASI \(PCI Geomatica\). Часть 2. Анализ данных](#)
- [Основы макроязыка EASI \(PCI Geomatica\). Часть 3. Географическая привязка и проекции](#)
- [Основы макроязыка EASI \(PCI Geomatica\). Часть 4. Примеры автоматизации процессов](#)

[Обсудить в форуме](#) Комментариев — 0

Последнее обновление: 2014-05-15 01:26

Дата создания: 15.08.2007

Автор(ы): Пётр Потапов