



# Руководство по Therion'у

**Stacho Mudrak**

**Martin Budaj**

Перевод – Горбенко Алексей ([http://vk.com/al\\_gorb](http://vk.com/al_gorb))



Therion – программное обеспечение, защищенное авторскими правами. Распространяется по лицензии GNU General Public License.

Copyright © 1999–2017 Stacho Mudrak, Martin Budaj.

В этой книге описывается Therion 5.4.1 (2017-04-18).

В разработке кода принимали участие *Olly Betts*, *Marco Corvi*, *Vladimir Georgiev*, *Georg Pacher* и *Dimitrios Zachariadis*.

Мы благодарны

*Martin Sluka*, *Ladislav Blazek*, *Martin Heller*, *Wookey*, *Olly Betts*

и всем пользователям за их замечания, поддержку и предложения.

Переводы (%):

Язык	XTherion	Карты	Loch	Переводчики
bg	86	87	100	Alexander Yanev, Ivo Tachev, Vladimir Georgiev
cz	81	88	–	Ladislav Blažek
de	82	92	–	Roger Schuster, Georg Pacher, Benedikt Hallinger
el	85	87	–	Stelios Zacharias
en[_GB _US]	75	93	100	Stacho Mudrák, Olly Betts
es	75	83	–	Roman Muñoz
fr	–	87	–	Eric Madelaine, Gilbert Fernandes
it	86	92	–	Marco Corvi
mi	–	91	–	Kyle Davis, Bruce Mutton
pl	–	90	–	Krzysztof Dudziński
pt[_BR _PT]	–	83	–	Toni Cavalheiro, Rodrigo Severo
ru	81	86	–	Василий Сухачев, Андрей Коженков
sk	85	93	96	Stacho Mudrák
sq	85	87	–	Fatos Katallozi
zh	86	91	–	Zhang Yuan Hai, Duncan Collis

На обложке изображен эскиз съемки грота *Hrozny kamenolom* в пещере Мертвых летучих мышей в Словакии и ее карта, созданная Therion'ом.

## **Содержание**



ПУСТЬ НЕ ВХОДИТ НИКТО, НЕ ЗНАЮЩИЙ ГЕОМЕТРИИ  
ΑΓΕΩΜΕΤΡΗΤΟΣ ΜΗΔΕΙΚ ΕΙΣΙΤΩ

— предполагаемая фраза над входом  
в Академию Платона, 4 век до н.э.

## Введение

Therion это программа для создания карт пещер. Ее цель состоит в том, чтобы помочь:

- архивировать данные съемки на компьютере в форме, максимально приближенной к оригинальным записям и зарисовкам, и обрабатывать их удобным и эффективным способом;
- рисовать красивые современные планы и разрезы;
- создавать реалистичные 3D-модели пещер.

Therion работает в операционных системах Unix, Linux, MacOS X и Win32. Исходный код и установщик для Windows доступны на веб-странице (<https://therion.speleo.sk>).

Therion распространяется под лицензией [GNU General Public License](#).

## Почему Therion?

В 1990-е мы активно занимались спелеологией и созданием карт. Имелось несколько компьютерных программ, которые строили нитку хода после закрытия колец и разброса ошибки. Это было большим подспорьем в работе, особенно работая над крупными и сложными пещерными системами. Мы использовали вывод одного из них (TJIKPR) в качестве фонового слоя со станциями для ручного рисования карт. После окончания огромного 166-страничного Атласа пещер мертвых летучих мышей в начале 1997 года у нас вскоре возникла проблема: мы нашли новые ходы, соединения между известными ходами. После обработки данных в TJIKPR, новые кольца повлияли на положение старых станций, большинство станций уже имели иную позицию из-за разброса невязки колец. Таким образом мы могли бы перерисовывать весь Атлас снова, или принять, что местоположение некоторых новых мест на карте было изображено не точно (в случае колец с длиной около 1 км ошибки достигали 10 м) и пытаться подогнать новые хода к старым съемкам.

Эти проблемы оставались, когда мы пытались рисовать карты с помощью некоторых программ CAD в 1998 и 1999 годах. Всегда было трудно добавить новые исследования без адаптации старых к новым рассчитанным позициям станций во всей пещере. Мы не нашли ни одной программы, которая могла бы нарисовать современную сложную карту (т.е. не только нитку хода с LRUD), в которых старые части съемки изменялись в соответствии с новыми расчетными координатами станций.

В 1999 году мы начали думать о создании собственной программы для рисования карт. Мы знали о программах, которые идеально подходили для конкретных подзадач. Это был METAPOST – язык программирования высокого уровня для описания векторной графики, Survex – отличная программа для обработки нитки хода, и TeX – для верстки результатов. Нужно было только сложить их вместе. В рождество 1999 года мы уже имели первую версию Therion'a. Она состояла примерно из 32 КБ Perl

скриптов и METAPOST макросов, но программа показала, что наши идеи были осуществимы.

В период 2000–2001 годов мы искали оптимальный формат входных данных, язык программирования, концепцию интерактивного редактирования карт и внутренних алгоритмов с помощью Martin Sluka (Прага) и Martin Heller (Zürich). В 2002 году мы представили первую версию Therion'a, которая отвечала нашим требованиям.

## Особенности

Therion – приложение для командной строки. Он обрабатывает входные файлы в текстовом формате, в том числе 2D-карты, и создает файлы с 2D-картами или 3D-моделью в качестве вывода.

Синтаксис входных файлов подробно описан в последующих главах. Вы можете создавать эти файлы в любом текстовом редакторе, например *ed* или *vi*. Файлы содержат инструкции для Therion'a, такие как:

`point 1303 1004 pillar`

где `point` – команда для символа точки, за которым следуют его координаты и специфический тип символа.

Ручное редактирование таких файлов не просто, особенно когда вы рисуете карты и вам нужно думать о пространственных (декартовых координатах). Поэтому существует специальный графический интерфейс для Therion'a, называемый XTherion. XTherion работает как расширенный текстовый редактор, редактор карт (где карты рисуются в полностью интерактивном режиме) и компилятор (который запускает Therion).

Это может выглядеть довольно сложно, но этот подход имеет много преимуществ:

- Строгое разделение данных и визуализации. В файлах данных указывается только то, что есть, а не то, на что это похоже. Визуальное представление добавляется METAPOST на более поздних этапах обработки данных (это очень похоже на представление XML-данных).

Это позволяет изменять символы карты, используемые без изменения входных данных, или объединить большие карты, созданные разными людьми в разных стилях, в одну карту с едиными символами.

2D-карты адаптированы для конкретного масштаба (уровень абстракции, нелинейное масштабирование символов и текстов).

- Все данные привязываются к положениям пикетов съемки. Если координаты станций съемки изменяются в процессе закрытия колец, то все связанные данные перемещаются соответственно, поэтому карта всегда актуальна.
- Therion не зависит от конкретной операционной системы, кодировки символов или редактора входных файлов; входные файлы останутся читабельными для человека.
- Можно добавить новые форматы вывода.
- 3D-модель создается из 2D-карт, чтобы получить реалистичную трехмерную модель не вводя слишком много данных.
- Хотя поддержка WYSIWYG ограничена, вы всегда можете получить то, что хотите.

## Требования к ПО

“Программа должна делать одну задачу, и должна делать это хорошо” (Кен Томпсон). Поэтому мы используем несколько внешних программ, которые обрабатывают и визуализируют данные. Therion в связке с другими программами может выполнить свою задачу намного лучше.

Therion'у необходимо:

- **TeX** дистрибутив. Необходимо только в том случае, если вы хотите создавать 2D-карты в формате PDF или SVG.
- **Tcl/Tk** с *BWidget* и опциональным расширением *tkImag*. Это требуется только для XTherion'a.
- **LCDF Typetools**, если вы хотите использовать легкую настройку для пользовательских шрифтов в PDF-картах.
- Утилиты *convert* и *identify* из дистрибутива ImageMagick, если вы хотите использовать деформирование эскизов.
- *ghostscript*, если вы хотите создавать калиброванные изображения с геопривязанными PDF-картами.

Установщик для Windows включает все необходимые пакеты, за исключением ghostscript. Прочтите *Приложение*, если вы хотите скомпилировать Therion самостоятельно.

Для отображения карт и моделей вы можете использовать любую из следующих программ:

- любой просмотрщик PDF или SVG для просмотра 2D-карт;
- любые GIS поддерживающие DXF или *shapefile* форматы для анализа карт;
- соответствующий 3D просмотрщик для моделей, экспортированных в формате отличном от стандартного;
- любой клиент базы данных SQL для обработки экспортированной базы данных.

## Инсталляция

### Установка из исходников (therion-5.\*.tar.gz package):

Исходники – главный дистрибутив Therion'a. Его необходимо скомпилировать и установить в соответствии с инструкциями в *Приложении*.

### Установка в Windows:

Запустите программу установки и следуйте инструкциям. Устанавливаются все необходимые материалы и создаются ярлыки для XTherion'a и Therion Book.

## Настройка среды

Therion считывает настройки из файла инициализации. Настройки по умолчанию должны работать отлично для пользователей использующих только латинские символы<sup>1</sup>, стандартные TeX и METAPOST.

---

<sup>1</sup> На PDF-картах Therion отображает большинство акцентированных символов как сочетание акцента и базового символа. Некоторые неявные акценты могут быть опущены. Предустановленные буквы с акцентом включены для словацкого и чешского языков.

Если вы хотите использовать собственные шрифты для латинских или не латинских символов в PDF-картах, отредактируйте файл инициализации. Инструкции о том, как это сделать приведены в *Приложении*.

### Как это работает?

Итак, теперь ясно, что нужно Therion'у, давайте посмотрим как он взаимодействует со всеми этими программами:



НЕ ПАНИКУЙТЕ! Когда ваша система настроена правильно, большинство из файлов скрыто от пользователя, и все необходимые программы автоматически запускаются Therion'ом.

Для работы с Therion'ом достаточно знать, что вам нужно создавать входные данные (лучше всего делать это в XTherion'e), запускать Therion и отображать выходные файлы (3D-модель, карта, лог-файл) в соответствующей программе.

Для тех, кто хочет больше узнать об этом, кратко изложим приведенную выше блок-схему. Названия программ отображены прямым шрифтом, а файлы данных выделены курсивом. Стрелки показывают поток данных между программами. Временные файлы данных не показаны. Значения цветов:

- черный – программы и макросы Therion'a (XTherion написан на Tcl/Tk, поэтому для него требуется этот интерпретатор);
- красный – T<sub>E</sub>X пакет;
- зеленый – входные файлы, созданные пользователем и выходные файлы, созданные Therion'ом.

Сам Therion выполняет главную задачу. Он считывает входные файлы, интерпретирует их, находит замкнутые кольца и раскидывает ошибки. Затем он преобразует все другие данные (например 2D-карты) в соответствии с позицией новых станций. Therion экспортирует данные для 2D-карт в формате METAPOST. METAPOST дает фактическую форму абстрактным символам карты в соответствии с определениями символов карты; он создает много файлов PostScript с небольшими фрагментами пещеры. Они считываются и преобразуются в PDF формат, который



формирует входные данные для pdfTeX. PdfTeX собирает все фрагменты и создает PDF-файл карты пещеры.

Therion также экспортирует трехмерную модель (полную или нитку хода) в различных форматах.

Нитка хода может быть экспортирована для дальнейшей обработки в любую базу данных SQL.

## Первый запуск

После объяснения основных принципов работы Therion'a давайте попробуем его на примерах реальных данных.

- Скачайте примеры данных с сайта Therion'a и распакуйте их на жесткий диск.
- Запустите XTherion (под Unix и MacOS X введя в командной строке 'xtherion', под Windows ярлык в меню *Пуск*). Откройте файл 'thconfig' из каталога примеров данных в окне 'therion компилятор'.
- Нажмите 'F9' или 'Компилировать' в меню для запуска Therion'a – вы получите несколько сообщений от Therion'a, METAPOST и TEX. PDF-карты и 3D-модель создаются в каталоге с данными.

Кроме того, вы можете открыть файлы данных съемки (\*.th) в окне 'therion текстовый редактор' и файлы абрисов карт (\*.th2) в окне 'therion редактор карт'. Наличие различных форматов данных может выглядеть запутанным по началу, но все они будут разъяснены в следующих главах.

Только для вас, гетей доктрины и обучения, мы написали эту работу. Изучите эту книгу, подумайте о том, что мы рассеяли в разных местах и собрались снова; что мы скрывали в одном месте, которое мы раскрыли в другом, что оно может быть понято вашей мудростью.

Vos igitur doctrinę & sapientię filii, perquirite in hoc libro, colligendo nostram disperſam intentionē, quam in diuerſis locis propoſuimus, & quod occultatum eſt à nobis in uno loco, manuſcriptum fecimus illud in alio, ut ſapientibus uobis pateſcat, uobis enim ſolis ſcripſimus

– Хенрикус С. Агриппа Ноттенсгеймский, 1533

## Создание файлов данных

### Основы

Входные файлы для Therion'a имеют текстовый формат. Существует несколько правил о том, как должен выглядеть такой файл:

- Есть два типа команд. Однострочные команды и многострочные команды.
- Однострочная команда завершается символом конца строки. Их синтаксис `command arg1 ... argN [-опция1 значение1 -опция2 значение2 ...]`

где `arg1 ... argN` являются обязательными аргументами, а пары `-опция значение` являются параметрами, которые вы можете опустить. Какие аргументы и опции доступны, зависит от конкретной команды. Примером может служить

`point 643.5 505.0 gradient -orientation 144.7`

с тремя обязательными аргументами и одной дополнительной парой опция/значение. Иногда параметров нет или может быть несколько значений.

- Многострочные команды начинаются аналогично однострочным, но продолжаются на последующих строках до явного завершения команды. Эти строки могут содержать либо данные, либо параметры, которые применяются к последующим данным. Если строка данных начинается со слова, зарезервированного для опции, вам нужно вставить '!' перед ней. Синтаксис:

`command arg1 ... argN [-опция1 значение1 -опция2 значение2 ...]`

...  
`опцияX значениеX`  
`данные`

...  
`endcommand`

Опять же, для лучшей иллюстрации приведем пример:

`line wall -id walltobereferenced`  
`1174.0 744.5`  
`1194.0 756.5 1192.5 757.5 1176.0 791.0`  
`smooth off`  
`1205.5 788.0 1195.5 832.5 1173.5 879.0`  
`endline`

Эта команда `line` имеет один обязательный аргумент, тип линии (коренная стена в данном случае), за которой следует одна опция. Следующие две строки содержат данные (координаты кривых Безье). Следующая строка ("`smooth off`") указывает параметр, который применяется к последующим

данным (т.е. не для всей линии, в отличие от опции `-id` в первой строке), и последняя строка содержит еще несколько данных.

- Если значение параметра или аргумента содержит пробелы, вы должны заключить это значение в `" "` или `[ ]`. Если вы хотите поместить двойную кавычку `"` в текст в `" "` вам нужно вставить его дважды. Кавычки используются для строк; скобки для числовых значений и ключевых слов.
- Каждая строка, заканчивающаяся обратным слэшем (`\`), считается продолженной на следующей строке, как будто не было ни разрыва строки, ни пробела.
- Все, что следует за `#` и до конца строки, даже внутри команды, считается комментарием и игнорируется.
- Многострочные комментарии внутри `comment ... endcomment` блока разрешены в файлах данных и конфигурационных файлах. 5.4

## Типы данных

Therion использует следующие типы данных:

- *ключевое слово* ▷ последовательность `A-Z`, `a-z`, `0-9` и `_`/`-` символов (не начинающиеся с `'-'`).
- *внешнее ключевое слово* ▷ слово, которое также может содержать `+`, `.`, `'` символы, но не в первой позиции.
- *дата* ▷ спецификация даты (или временного интервала) в формате `YYYY.MM.DD@HH:MM:SS.SS - YYYY.MM.DD@HH:MM:SS.SS` или `'-'` чтобы указать неопределенную дату.
- *персона* ▷ имя и фамилия человека, разделенные пробельными символами. Используйте `'/'` чтобы отделить имя и фамилию, если есть несколько имен.
- *строка* ▷ последовательность любых символов. Строки могут содержать специальный тег `<lang:XX>` для разделения переводов. В многоязычных строках только текст между `<lang:XX>` (где `XX` – это язык, выбранный в файле инициализации или конфигурации) и следующим тегом `<lang:YY>` отображается на выходе. Если совпадение не найдено, все до появления тега `<lang:ZZ>` отображается. 5.3
- *единицы* ▷ поддерживаемые единицы длины: `meter[s]` (метры), `centimeter[s]` (сантиметры), `inch[es]` (дюймы), `feet[s]` (футы), `yard[s]` (ярды) (можно сокращать `m`, `cm`, `in`, `ft`, `yd`). Поддерживаемые угловые единицы: `degree[s]` (градусы), `minute[s]` (минуты) (можно сокращать `deg`, `min`), `grad[s]` (грады), `mil[s]` (тысячные), `percent[age]` (проценты, только для угла наклона). Значение градуса может быть введено в десятичной системе (`x.y`) или в специальной нотации для градусов, минут и секунд (`deg[:min[:sec]]`).

## Системы координат

Therion поддерживает преобразования координат в геодезические системы координат. Вы можете указать опцию `cs` в объектах `centreline`, `surface`, `import` и `layout` и ввести XY в выбранной системе координат. Вы также можете указать вывод `cs` в конфигурационном файле.

Если вы не указали какой-либо `cs` в вашем наборе данных, то предполагается, что вы работаете в `local` системе координат, и никакие преобразования не выполняются. Если вы укажете `cs` в любом месте данных, то вы должны указать его для всех данных местоположения (`fix`, `origin` и `layout` и т.д.).

Cs применяется ко всем последующим данным местоположения, пока другие cs не будут указаны или до конца текущего объекта, в зависимости от того, что наступит раньше.

Поддерживаются следующие системы координат:

- **UTM1 – UTM60** ▷ Универсальная поперечная проекция Меркатора (Universal Transverse Mercator) в северном полушарии и заданной зоне, WGS84.
- **UTM1N – UTM60N** ▷ То же, что и **UTM1 – UTM60**.
- **UTM1S – UTM60S** ▷ UTM в южном полушарии, WGS84.
- **lat-long, long-lat** ▷ Широта (N положительная, S отрицательная) и долгота (E положительная, W отрицательная) в заданном порядке в градусах (разрешено `deg[:min[:sec]]`), WGS84. По умолчанию не поддерживается на выходе.
- **EPSG:<число>** ▷ Большинство систем координат EPSG. Почти каждая система координат, используемая во всем мире, имеет собственный номер EPSG. Чтобы найти номер вашей системы, смотрите файл [extern/proj4/nad/epsg](#) в дистрибутиве исходников.
- **ESRI:<число>** ▷ Аналогично EPSG, но стандарт ESRI.
- **JTSK, iJTSK** ▷ Чехословацкая система S-JTSK, используемая с 1920-х годов с южной и западной осью (JTSK) и ее модифицированной версией с осью, указывающей восток и север отрицательными числами (iJTSK). JTSK не поддерживается на выходе (как и iJTSK).
- **JTSK03, iJTSK03** ▷ Новая реализация S-JTSK, введенная в Словакии в 2011 году.
- **OSGB:<N, N, O, S или T><A-Z исключая I>** ▷ Британская Национальная Сеть.
- **S-MERC** ▷ Сферическая проекция Меркатора, используемая различными сайтами онлайн-сопоставления.

5.4

## Магнитное склонение

Therion содержит встроенный IGRF<sup>2</sup> – модель геомагнитного поля Земли, действительная для периода 1900–2020 гг. Она автоматически используется, если пещера находится в пространстве с использованием любой из поддерживаемых геодезических систем координат, и никакое склонение не определяется пользователем. Вычисленное склонение указано для информации в файле LOG.

5.4

## Формат данных

Синтаксис входных файлов объясняется в описании отдельных команд. Изучение примеров файлов, распространяемых вместе с Therion'ом, поможет вам понять основы. Смотрите также примеры в *Приложении*.

В каждом из следующих разделов описывается одна команда Therion'а, использующая следующую структуру:

*Описание:* примечания относительно этой команды.

*Синтаксис:* описание синтаксиса.

---

<sup>2</sup> Смотрите <https://www.ngdc.noaa.gov/IAGA/vmod/>

*Контекст:* указывает контекст, в котором используется эта команда. Контекст *survey* означает, что команда должна быть заключена в пару *survey ... endsurvey*. Контекст *scrap* означает, что команда должна быть заключена в пару *scrap ... endscrap*. Контекст *all* означает, что команда может использоваться в любом месте.

*Аргументы:* список обязательных аргументов с пояснениями.

*Опции:* список доступных опций.

*Опции командной строки:* для многострочных команд, которые могут быть указаны среди строк данных.

## **'encoding'**

*Описание:*

Устанавливает кодировку входного файла. Это позволяет использовать символы не ASCII во входных файлах.

*Синтаксис:*

*encoding* <имя кодировки>

*Контекст:*

Это должна быть самая первая команда в файле.

*Аргументы:*

- <имя кодировки> ▷ чтобы увидеть список всех поддерживаемых имен кодировок, запустите Therion с опцией *-print-encodings*. Кодировки 'UTF-8' (Unicode) и 'ASCII' (7 бит) всегда поддерживаются.

## **'input'**

*Описание:*

Вставляет содержимое файла на место команды. Расширение по умолчанию '.th' и его можно не указывать. Для максимальной портатбельности используйте относительные пути, а для разделения каталогов используйте характерный для Unix '/', а не обратный слеш используемый в Windows '\\'.  
*Синтаксис:*

*input* <имя файла>

*Контекст:*

all

*Аргументы:*

- <имя файла>

## **'survey'**

*Описание:*

Survey – основная структура данных.

Survey могут быть вложенными – это позволяет построить иерархическую структуру. Обычный уровень иерархической структуры survey представлен пещерами, более высокие уровни карстовыми областями, а более низкие уровни, например, ходами пещеры.

Каждый survey имеет собственное пространство имен, указанный его аргументом `<идентификатор>`. Объекты (например станции или скрапы, смотрите ниже), которые относятся к subsurvey текущего survey, записываются как

`<идентификатор объекта>@<идентификатор subsurvey>`,

или, если есть больше уровней вложения

`<идентификатор объекта>@<идентификатор subsubsurvey>`.

`<идентификатор subsurvey>`<sup>3</sup>.

Это означает, что идентификаторы объектов должны быть уникальными только в рамках одного survey. Например, имена станций съемки могут быть одинаковыми, если они находятся в разных survey. Это позволяет делать нумерацию станций с 0 в каждом survey или объединение двух пещер в одну пещерную систему без переименования станций съемки.

*Синтаксис:*

`survey <идентификатор> [ОПЦИИ]`

... Другие объекты ...

`endsurvey [<идентификатор>]`

*Контекст:*

none, survey

*Аргументы:*

- `<идентификатор>` ▷ идентификатор.

*Опции:*

- `namespace <on/off>` ▷ указывает, создавать ли survey пространство имен (on по умолчанию).
- `declination <спецификация>` ▷ устанавливает склонение по умолчанию для всех объектов данных в этом survey (которые могут быть переопределены новым склонением в subsurveys). `<Спецификация>` имеет три формы:

1. `[]` пустую строку. Это приведет к сбросу склонения.
2. `[<значение> <единицы>]` установит одно значение (также для пустого survey).
3. `[<дата1> <значение1> [<дата2> <значение2> ... ] <единицы>]` установит склонение для нескольких дат. Тогда склонение каждого замера будет установлено в соответствии со спецификацией даты объекта данных. Если вы хотите явно указать склонение для данных без даты, используйте '-' вместо даты.

Если не определено склонение и определена какая-либо геодезическая система координат, склонение автоматически вычисляется с использованием встроенной геомагнитной модели.

Обратите внимание: склонение положительно, когда магнитный север находится к востоку от истинного севера.

- `person-rename <старое имя> <новое имя>` ▷ переименовать человека, имя которого было изменено.
- `title <строка>` ▷ описание объекта.
- `entrance <имя-станции>` ▷ указывает главный вход в пещеру, представленный в этом survey. Если это не указано, и в этом survey есть только одна станция отмеченная входом, он считается также входом в пещеру. Эта информация используется для `cave-list` экспорта.

---

<sup>3</sup> Примечание: невозможно связать любой объект с survey более высокого уровня.

## **'centreline'**

### *Описание:*

Описание данных survey (нитки хода). Синтаксис заимствован из Survox с небольшими изменениями; руководство Survox может быть полезно в качестве дополнительной справки для использования. Можно использовать синоним 'centerline'.

### *Синтаксис:*

```
centreline [ОПЦИИ]
  date <дата>
  team <персона> [<роли>]
  explo-date <дата>
  explo-team <персона>
  instrument <прибор> <описание>
  calibrate <параметр> <ошибка> [<масштаб>]
  units <параметр> [<фактор>] <единицы>
  sd <параметр> <значение> <единицы>
  grade <grade list>
  declination <значение> <единицы>
  grid-angle <значение> <единицы>
  infer <что> <on/off>
  mark <тип>
  flags <флаги замеров>
  station <станция> <комментарий> [<флаги>]
  cs <система координат>
  fix <станция> [<x> <y> <z> [<std x> <std y> <std z>]]
  equate <список станций>
  data <стиль> <порядок считывания>
  break
  group
  endgroup
  walls <auto/on/off>
  vthreshold <число> <единицы>
  extend <спес> [<станция> [<станция>]]
  station-names <префикс> <суффикс>
  ...
  [данные съемки]
  ...
endcentreline
```

### *Контекст:*

none, survey

### *Опции:*

- **id** <идентификатор> ▷ идентификатор объекта.
- **author** <дата> <персона> ▷ автор данных и дата их создания.
- **copyright** <дата> <строка> ▷ дата и имя авторского права.
- **title** <строка> ▷ описание объекта.



### Опции командной строки:

- **date** <дата> ▷ дата съемки. Если указано несколько дат, создается временной интервал.
- **explo-date** <дата> ▷ дата исследования. Если указано несколько дат, создается временной интервал.
- **team** <персона> [<роли>] ▷ член съемочного отделения. Первый аргумент – его имя, остальные описывают роли человека в команде (необязательно – в настоящее время не используется). Поддерживаются следующие зарезервированные слова: station (расстановка пикетов), length (измерение расстояния), tape (рулетка), [back]compass (компас), [back]bearing (азимут), [back]clino (угол наклона), [back]gradient (уклон), counter (счетчик), depth (глубина), position (позиция), notes (записи), pictures (изображения), pics (изображения), instruments (insts) (инструменты), assistant (dog) (помощник).
- **explo-team** <персона> ▷ член исследовательского отделения.
- **instrument** <инструмент> <описание> ▷ описание инструмента, который использовался для определения количественных данных (те же значения, что и роль человека в команде).
- **infer** <что> <on/off> ▷ 'infer plumbs on' говорит программе интерпретировать угол наклона  $\pm 90^\circ$  как UP/DOWN (это означает, что угол наклона не будет изменяться при раскидывании невязки). 'infer equates on' программа будет интерпретировать замеры с длиной 0 в качестве команды equate (что означает, что не применяются никакие корректировки длины).
- **declination** <значение> <единицы> ▷ задает склонение для последующих замеров

истинный азимут = измеренный азимут + склонение

Склонение положительно, когда магнитный север находится к востоку от истинного севера. Если ни одно склонение не указано, или склонение сброшено (-), тогда действительная спецификация склонения ищется во всех съемках, в которых находится объект данных. Смотрите опцию declination команды survey.

- **grid-angle** <значение> <единицы> ▷ задает угол магнитной сетки (склонение от направления на север).
- **sd** <параметр> <значение> <единицы> ▷ задает стандартное отклонение для данных измерений. <Параметр> может содержать следующие значения: length, tape, bearing, compass, gradient, clino, counter, depth, x, y, z, position, easting, dx, northing, dy, altitude, dz.
- **grade** <grade list> ▷ устанавливает стандартные отклонения для угла наклона съемки (смотрите команду grade). Все ранее заданные стандартные отклонения или определения теряются. Если вы хотите изменить SD, используйте опцию sd после этой команды. Если указано несколько определений, то применяется только последнее. Вы можете указать определения только для позиции или только для съемок. Если вы хотите их комбинировать, вы должны использовать их в одной строке.
- **units** <параметр> [<фактор>] <единицы> ▷ устанавливает единицы для снимаемых параметров (так же как и для sd).
- **calibrate** <параметр> <ошибка> [<масштаб>] ▷ установка калибровки прибора. Измеренное значение рассчитывается по следующей формуле:



измеренное значение = (прочитанное значение – ошибка) × масштаб.  
Поддерживаемые единицы такие же, как в sd.

- **break** ▷ может использоваться между данными, чтобы отделить два пересечения.
- **mark** [**<список станций>**] **<тип>** ▷ установить тип именованных станций. **<тип>** может принимать одно из значений: **fixed** (фиксированным), **painted** (нарисованный) и **temporary** (временным) (по умолчанию). Если список станций отсутствует, все последующие станции будут отмечены соответствующим типом.
- **flags** **<флаги замеров>** ▷ установить флаги для следующих замеров. Поддерживаемые флаги: **surface** (для измерений на поверхности), **duplicate** (для дублированных съемок), **splay** (для коротких боковых замеров, которые по умолчанию скрыты на картах и моделях). Они исключаются из расчетов длины.

Все замеры, имеющие в названии одной из станций '.' или '-' по умолчанию имеют тип **splay** (смотрите также команду **data**).

Если флаг установлен на **approx[imate]**, он включается в вычисления общей длины, но также отображается отдельно в статистике съемки. Он должен использоваться для замеров, которые не были сняты должным образом и нуждаются в пересъемке.

Также перед флагом допускается приставка **"not"**.

- **station** **<станция>** **<комментарий>** [**<флаги>**] ▷ установить комментарий станции и ее флаги. Если в качестве комментария указан "", то он игнорируется.

Поддерживаемые флаги: **entrance** (вход), **continuation** (продолжение), **air-draught[:winter/summer]** (ток воздуха:зима/лето), **sink** (пonor), **spring** (источник), **doline** (карстовая воронка), **dig** (раскоп), **arch** (свод), **overhang** (нависание потолка). Также приставка **not** может быть использована перед флагом, чтобы удалить ранее добавленный флаг.

5.3

Вы также можете указать пользовательские атрибуты для станции, используя флаг **attr**, за которым следуют имя и значение атрибута.  
Пример:

**station 4 "колодец для изучения" continuation attr code "V"**

Если есть ход, который был исследован, но еще не снят, то предполагаемая исследуемая длина этого хода может быть учтена добавлением к станции флага **continuation**. Просто добавьте флаг **explored** **<исследовательская длина>** для станции. Исследовательские длины являются частью статистики съемки/пещеры, отображаемой отдельно. Пример:

**station 40 "ужасная прогулка" continuation explored 100m**

- **cs** **<система координат>** ▷ система координат для станций с фиксированными координатами.
- **fix** **<станция>** [**<x>** **<y>** **<z>**] [**<std x>** **<std y>** **<std z>**] ▷ фиксировать координаты станции (с указанными ошибками – к ним применяется только преобразование единиц, а не калибровка).
- **equate** **<список станций>** ▷ эквивалентность заданных точек.
- **data** **<стиль>** **<порядок считывания>** ▷ установить стиль данных (**normal**, **topofil**, **diving**, **cartesian**, **cylopolar**, **dimensions**, **nosurvey**) и порядок считывания. Считывание может принимать одно из следующих ключевых слов: **station** (станция), **from** (начальная станция), **to** (конечная станция), **tape/length** (длина), **[back]compass/[back]bearing** (азимут),

[back]clino/[back]gradient (угол наклона), depth (глубина), fromdepth (от глубины), todepth (до глубины), depthchange (изменение глубины), counter (счетчик), fromcount (от счетчика), tocount (к счетчику), northing (север), easting (восток), altitude (высотная отметка), up/ceiling (высота потолка)<sup>4</sup>, down/floor, left, right, ignore, ignoreall.

Смотрите руководство к Survex для подробностей.

Для разноуровневых данных поддерживаются ключевые слова новой строки и направления. Если есть прямые и обратные данные азимута или угла, то вычисляется среднее из них.

5.3

Если в замере одна из станций имеет название '.' или '-', то у замера устанавливается атрибут **splay**. Точку следует использовать для замеров, заканчивающихся внутри хода, *type* для замеров, заканчивающихся на стенах хода, на полу или потолке. Хотя Therion еще не делает различий между ними, его следует использовать для улучшения 3D-моделирования в будущем.

- **group**
- **endgroup** ▷ **group/endgroup** пара позволяет пользователю делать временные изменения практически в любых настройках (калибровка, единицы, sd, данные, флаги...).
- **walls <auto/on/off>** ▷ включение/выключение генерации формы хода из данных LRUD для последующих замеров. Если установлен **auto**, то ход генерируется только в том случае, если отсутствует скрап содержащий данную нитку хода.
- **vthreshold <число> <единицы>** ▷ пороговое значение для интерпретации показаний LRUD как показания лево-право-верх-низ, перпендикулярное замеру.

Если ходы горизонтальны (**наклон** < **vthreshold**), LR считается перпендикулярным к замеру, а UD вертикальными.

Если ходы более или менее вертикальные (**наклон** > **vthreshold**), тогда даже UD становится перпендикулярным к замеру, в противном случае ходы выглядят не очень хорошо. В случае вертикальных замеров UD интерпретируется как измерение с севера на юг от станции, чтобы построить вертикальные участки модели.

- **extend <spec> [<станция> [<станция>]]** ▷ определение развертки нитки хода. **<spec>** принимает следующие значения:

**normal/reverse** ▷ развернуть данную и следующие станции в таком же/обратном направлении по отношению к предыдущей станции. Если заданы две станции – направление применяется только к данному замеру;

**left/right** ▷ как указано выше, но направление указано явно;

**vertical** ▷ не перемещает станцию (замер) в направлении X, использует только Z компонент замера;

**start** ▷ указать начальную станцию (замер);

**ignore** ▷ игнорирует указанную станцию (замер), продолжает развертку как у другой станции (замера), если возможно;

---

<sup>4</sup> Размер может быть задан как пара [**<from>** **<to>**], что означает размер в начале и в конце замера.

`hide` ▷ не показывает указанную станцию (замер) в развертке.

Если ни одна из станций не указана, `<spec>` действует для следующих указанных замеров.

- `station-names` `<префикс>` `<суффикс>` ▷ добавляет данный префикс/суффикс для всех станций съемки в текущей нитки хода. Для сохранения типизации.

## 'scrap'

Описание:

Скрап – это часть 2D-карты, которая не содержит пересекающихся ходов (т.е. все хода могут быть нарисованы на бумаге не пересекаясь). Для небольших и простых пещер вся пещера может поместиться в одном скрапе. В сложных системах скрап обычно представляет собой один грот или один ход. В идеале скрап содержит около 100 м пещеры<sup>5</sup>. Каждый скрап обрабатывается отдельно METAPOST; слишком большие скрапы могут превысить размер памяти METAPOST и вызвать ошибки.

Скрап состоит из точечных, строковых и областных символов. Смотрите главу *Как карта собирается в единое целое* для объяснения, как и в каком порядке они выводятся при отображении.

Граница скрапа состоит из линий с опцией `-outline out` или `-outline in` (стены хода по умолчанию имеют опцию `-outline out`). Эти линии не должны пересекаться, иначе, Therion (METAPOST) не может определить внутреннюю часть скрапа и METAPOST выдает предупреждение "`scrap outline intersects itself`".

У каждого скрапа есть своя собственная локальная декартова система координат, которая обычно соответствует миллиметровой бумаге (если Вы определяете координаты символов карты вручную) или пикселям отсканированного изображения (если Вы используете XTherion). Therion делает преобразование этой локальной системы координат в реальные координаты, используя позиции станций съемки, которые определены в скрапе как символы точки на карте и определены в нитке хода. Если скрап не содержит по крайней мере две станции съемки с определением `-name`, тогда Вы должны использовать опцию `-scale` для калибровки скрапа (это обычно используется для сечений).

Трансформация состоит из следующих шагов:

- линейное преобразование (смещение, масштабирование и вращение), которые 'лучше всего' соответствуют станциям. 'Лучше всего' означает, что сумма квадратов расстояний между соответствующими станциями до и после трансформации минимальна. Результат отображается красным цветом, если опция `debug` в команде `layout` установлена в `on`.

---

<sup>5</sup> Если необходимо, скрапы могут быть намного меньше, несколько метров пещеры. При определении размера скрапа учитывайте следующее: использование небольших скрапов может потребовать больше времени для обрисовки и для оптимизации их объединения. С другой стороны, маленькие скрапы, вероятно, будут менее искажены алгоритмами преобразования карт, чем более крупные скрапы. Слишком большие скрапы могут исчерпать память METAPOST, если часто используются заливки хода. Кроме того, редактор карт в XTherion'e медленнее реагирует при редактировании больших скрапов.

- Нелинейное преобразование скрапа состоит из перемещения станций съемки в правильное положение. Отображается синим цветом в режиме `debug`.
- Нелинейное преобразование скрапа, которое состоит из (1) перемещения соединенных вместе точек, (2) не перемещает станции съемки. В конце положение контрольных точек кривых корректируется, чтобы сохранить гладкость. Результат – готовая карта.

*Синтаксис:*

```
scrap <идентификатор> [ОПЦИИ]
... команды точек, линий и областей ...
endscrap [<идентификатор>]
```

*Контекст:*

none, survey

*Аргументы:*

- `<идентификатор>` ▷ идентификатор скрапа.

*Опции:*

- `projection <спецификация>` ▷ указывает проекцию чертежа. Каждая проекция идентифицируется типом и дополнительно индексом в форме `тип[:индекс]`. Индекс может быть любым ключевым словом. Поддерживаются следующие типы проекций:
  1. `none` ▷ нет проекции, используется для сечений или карт, которые не зависят от данных съемки (например, оцифровка старых карт, где нет нитки хода). Для этой проекции не указывается индекс.
  2. `plan` ▷ проекция основного плана (по умолчанию).
  3. `elevation` ▷ ортогональная проекция (проецируемый профиль a.k.a.), которая опционально принимает в качестве аргумента азимут сечения (например, `[elevation 10]` или `[elevation 10 deg]`).
  4. `extended` ▷ разрез-развертка (развертка a.k.a.).
- `scale <спецификация>` ▷ используется для предварительного масштабирования (преобразования координат пикселей в метры) данных скрапа. Если проекция скрапа установлена в `none`, то это единственное преобразование, которое выполняется с координатами. `<Спецификация>` имеет четыре формы:
  1. `<число>` ▷ `<число>` метров на единицу чертежа.
  2. `[<число> <единица длины>]` ▷ `<число>` `<единиц длины>` на единицу чертежа.
  3. `[<число1> <число2> <единица длины>]` ▷ `<число1>` чертежных единиц соответствует `<число2>` `<единиц длины>` в реальности.
  4. `[<число1> ... <число8> [<единица длины>]]` ▷ это самый общий формат, где вы указываете по порядку координаты  $x$  и  $y$  двух точек в скрапе и две точки в реальности. При желании вы также можете указать единицы для координат 'точек в реальности'. Эта форма позволяет применять как масштабирование, так и поворот скрапа.
- `cs <система координат>` ▷ предполагается, что (калиброванные) локальные координаты скрапа приведены к заданной системе координат. Это полезно

для абсолютного размещения импортированных эскизов, где не указана ни одна станция съемки<sup>6</sup>.

- `stations <список имен станций>` ▷ станции, которые вы хотите использовать в скрапе, но которые не используются для его трансформации. Вам не нужно указывать (рисовать) их с помощью команды `point station`.
- `sketch <имя файла> <x> <y>` ▷ определение растрового рисунка в качестве эскиза/подложки (координаты нижнего левого угла).
- `walls <on/off/auto>` ▷ указывает, следует ли использовать скрап в создании 3D-модели.
- `flip (none)/horizontal/vertical` ▷ зеркально отображает скрап после трансформации масштаба.
- `station-names <префикс> <суффикс>` ▷ добавляет префикс/суффикс всем станциям скрапа в текущей съемки. Для сохранения типизации.
- `author <дата> <персона>` ▷ автор данных и дата их создания.
- `copyright <дата> <текст>` ▷ авторские права и их дата.
- `title <текст>` ▷ описание объекта.

## 'point'

*Описание:*

Точка – это команда для рисования символа точки.

*Синтаксис:*

`point <x> <y> <тип> [ОПЦИИ]`

*Контекст:*

scrap

*Аргументы:*

- `<x>` и `<y>` являются координатами чертежа объекта.
- `<тип>` определяет тип объекта. Поддерживаются следующие типы:  
*специальные объекты:* `station`<sup>7</sup> (станция), `section`<sup>8</sup> (сечение), `dimensions`<sup>9</sup> (размеры хода);  
*метки:* `label` (текстовая метка), `remark` (заметка), `altitude`<sup>10</sup> (высотная отметка), `height`<sup>11</sup> (высота уступа/камина/колодца), `passage-height`<sup>12</sup>

---

<sup>6</sup> Если в скрапе есть несколько станций съемки, `cs` игнорируется.

<sup>7</sup> Станция съемки. Для каждого скрапа (за исключением скрапов с проекцией 'none') должна быть указана хотя бы одна станция с опцией `-name`.

<sup>8</sup> `Section` является местом для размещения поперечного сечения в этой точке. Этот символ не имеет визуального представления. Поперечное сечение должно быть в отдельном скрапе с проекцией 'none'. Вы можете указать ее через опцию `-scrap`.

<sup>9</sup> Используя опцию `-value` можно указать размеры хода выше/ниже плоскости нитки хода, используемой при создании 3D-модели.

<sup>10</sup> Знак общей высоты. Все высоты экспортируются как разница от *Z* сетки (по умолчанию 0). Чтобы отобразить высоту на стене прохода, используйте опцию `altitude` для любой точки линии стены хода.

<sup>11</sup> Высота внутри хода (например, яма и т.д.); смотрите ниже подробнее.

<sup>12</sup> Высота хода; смотрите ниже.

(высота хода), `station-name`<sup>13</sup> (имя станции, номер пикета), `date` (дата наблюдения);

*символы заполнения хода*<sup>14</sup>: `bedrock` (коренная порода), `sand` (песок), `raft` (конус), `clay` (глина), `pebbles` (галька), `debris` (щебень), `blocks` (навал глыб), `water` (вода), `ice` (лед), `guano` (гуано), `snow` (снег);

*спелео-формы*: `flowstone` (натечный каскад), `moonmilk` (мондмилх), `stalactite` (сталактиты), `stalagmite` (сталагмиты), `pillar` (колонны, сталагматы), `curtain` (натечный занавес), `helictite` (геликтиты), `soda-straw` (сталактиты-соломины), `crystal` (кристаллы), `wall-calcite` (настенный кальцит), `popcorn` (коралиты), `disk` (диск), `gypsum` (гипс), `gypsum-flower` (гипсовые цветы), `aragonite` (арагонит), `cave-pearl` (пещерный жемчуг), `rimstone-pool` (гуры с водой), `rimstone-dam` (гуры без воды), `anastomosis` (анастомозис), `karren` (карры), `scallop` (фасетки), `flute` (канавки), `raft-cone` (высыпной конус), `clay-tree` (глиняное дерево);

*оборудование*: `anchor` (ИТО), `rope` (веревка), `fixed-ladder` (жесткая лестница), `rope-ladder` (гибкая лестница), `steps` (ступени), `bridge` (мост), `traverse` (троллей), `camp` (ПБЛ), `no-equipment` (снаряжение отсутствует);

*окончание ходов*: `continuation` (продолжение, возможное продолжение), `narrow-end` (вертикальная узость), `low-end` (горизонтальная узость), `flowstone-choke` (блокировано натеками), `breakdown-choke` (блокировано обвалом), `clay-choke` (блокировано глиной), `entrance` (вход);

*группы*: `dig` (раскоп), `archeo-material` (археологическая находка), `paleo-material` (палео-находка), `vegetable-debris` (останки растительности), `root` (корни растений), `water-flow` (водоток), `spring` (источник)<sup>15</sup>, `sink` (попор), `ice-stalactite` (ледяной сталактит), `ice-stalagmite` (ледяной сталагмит), `ice-pillar` (ледяная колонна, ледяной сталагнат), `gradient` (уклон хода), `air-draught` (ток воздуха)<sup>16</sup>, `map-connection` (соединение выносок карт)<sup>17</sup>, `extra` (точка морфинга), `u`<sup>18</sup> (пользовательский).

*Опции:*

- `subtype` <подтип> ▷ определяет подтип объекта. Поддерживаются следующие подтипы для заданных типов:

*station* (станция)<sup>19</sup>: `temporary` (временная) (по умолчанию), `painted` (нарисованная), `natural` (естественная), `fixed` (фиксированная);

*air-draught* (ток воздуха): `winter` (зимой), `summer` (летом), `undefined` (не определено) (по умолчанию);

*water-flow* (водоток): `permanent` (постоянный) (по умолчанию), `intermittent` (временный), `paleo` (палео-поток).

<sup>13</sup> Если текст не указан, используется имя ближайшей станции.

<sup>14</sup> В отличие от других точечных символов, они подрезаются границей скрапа. Смотрите главу Как карта собирается в единое целое.

<sup>15</sup> Всегда используйте символы `spring` (источник) и `sink` (попор) со стрелкой `water-flow` (водоток).

<sup>16</sup> Количество штрихов на стрелке устанавливается в зависимости от опции `-scale` (масштаб). Примечание переводчика: по рекомендациям международной системы обозначений должно быть всегда два штриха.

<sup>17</sup> Виртуальная точка, используемая для указания соединения между выносками карт (разрез-развертка, смещение карты).

<sup>18</sup> Пользовательские точечные символы.

<sup>19</sup> Если подтип станции не указан, Therion считает ее пикетом, если она указана в нитке хода.



Подтип может быть указан также непосредственно в `<type>` с использованием ':' в качестве разделителя<sup>20</sup>.

Любой подтип может использоваться с пользовательским типом (u). В этом случае вам также необходимо определить соответствующий метапост символ (смотрите главу *Новые символы карты*).

- `orientation/orient <число>` ▷ определяет ориентацию символа. Если не указано, то ориентировано на север.  $0 \leq \text{число} < 360$ .
- `align` ▷ выравнивание символа или текста. Принимает следующие значения: center, c, top, t, bottom, b, left, l, right, r, top-left, tl, top-right, tr, bottom-left, bl, bottom-right, br.
- `scale` ▷ масштаб символа, может принимать значения: tiny (xs), small (s), normal (m), large (l), huge (xl) или числовое значение. По умолчанию normal. Именованные масштабы кратны  $\sqrt{2}$ , и имеют следующие значения:  $xs \equiv 0.5$ ,  $s \equiv 0.707$ ,  $m \equiv 1.0$ ,  $l \equiv 1.414$  и  $xl \equiv 2.0$ .
- `place <bottom/default/top>` ▷ порядок отображения изменений на карте.
- `clip <on/off>` ▷ указывает, подрезать ли символ границей скрапа. Вы не можете указать этот параметр для следующих символов: station (станция), station-name (имя станции, номер пикета), label (текстовая метка), remark (заметка), date (дата наблюдения), altitude (высотная отметка), height (высота уступа/камина/колодца), passage-height (высота хода).
- `dist <дистанция>` ▷ применяется для дополнительных точек, указывает расстояние до ближайшей станции (или станции, указанной с помощью опции -from. Если не указано, используется соответствующее значение из данных LRUD.
- `from <станция>` ▷ применяется для дополнительных точек, указывает опорную станцию.
- `visibility <on/off>` ▷ отображает/скрывает символ.
- `context <point/line/area> <тип символа>` ▷ (для использования с опциями `symbol-hide` (скрыть символ) и `symbol-show` (отобразить символ) в layout) символ будет скрыт/показан в соответствии с правилами для указанного <типа символа><sup>21</sup>.
- `id <идентификатор>` ▷ идентификатор символа.

*Специальные параметры:*

- `name <ссылка>` ▷ если тип точки – это station (станция), то эта опция дает ссылку на реальную станцию съемки.
- `extend [prev[ious] <станция>]` ▷ если тип точки – это station (станция), а скрап является extended elevation (разрез-разверткой), то вы можете регулировать развертку нитки хода, используя эту опцию.
- `scrap <сноска>` ▷ если тип точки – это section (сечение), то эта сноска ссылается на скрап поперечного сечения.
- `explored <длина>` ▷ если тип точки – это continuation (продолжение, возможное продолжение), то вы можете указать длину ходов,

---

<sup>20</sup> Например, `station:fixed`.

<sup>21</sup> Пример: если вы укажете `-context point air-draught` текстовую метку, на которой отображается дата наблюдения, команда `symbol-hide point air-draught` скроет стрелку ток воздуха и соответствующую ей текстовую метку.

исследованных, но пока не снятых. Это значение будет отображаться в статистике съёмки/пещеры.

- **text** ▷ текст label (текстовая метка), remark (замечание) или continuation (продолжение, возможное продолжение). Он может содержать следующие зарезервированные слова для форматирования<sup>22</sup>:

**<br>** ▷ разрыв строки;

**<center>/<centre>**, **<left>**, **<right>** ▷ выравнивание строк для многострочных меток. Игнорируется, если нет тега **<br>**;

**<thsp>** ▷ тонкий пробел;

**<rm>**, **<it>**, **<bf>**, **<ss>**, **<si>** ▷ переключают шрифты;

5.3 **<rtl>** и **</rtl>** ▷ отмечает начало и конец вывода текста справа налево;

5.3 **<lang:XX>** ▷ создает многоязычную текстовую метку (подробнее смотрите описание типа **string**).

- **value** ▷ значение точек height (высота уступа/камина/колодца), passage-height (высота хода), altitude (высотная отметка) или dimensions (размеры хода).

*height (высота уступа/камина/колодца)*: в зависимости от знака значения (положительный, отрицательный или без знака), этот тип символа представляет собой высоту трубы, глубину колодца или высоту уступа. Числовое значение может сопровождаться необязательным символом '?', чтобы добавить единицы измерения (например **-value [40? ft]**).

*passage-height (высота хода)*: поддерживаются следующие четыре формы значения: **+<число>** (высота потолка), **-<число>** (глубина этажа или глубина воды), **<число>** (расстояние между потолком и полом) и **[+<число> -<число>]** (расстояние до потолка и расстояние до пола).

*altitude (высотная отметка)*: указанное значение представляет собой разницу высот по сравнению с ближайшей станцией. Если значение altitude (высотная отметка) имеет префикс "fix" (например **-value [fix 1300]**), то это значение используется как абсолютная высота. Значение может сопровождаться единицами измерения.

*dimensions (размеры хода)*: **-value [<выше> <ниже> [<единицы>]]** определяет размеры ходов выше/ниже плоскости нитки хода, используется в 3D-модели.

## 'line'

*Описание:*

Line – это команда для рисования символа линии на карте. Каждый символ линии ориентирован, и его визуализация может зависеть от его ориентации (например, галочка на конце линии). Главное правило заключается в том, что свободное пространство хода находится слева по ходу линии (галочка направлена в эту сторону), а порога справа.

*Синтаксис:*

---

<sup>22</sup> Для вывода SVG учитываются только зарезервированные слова **<br>**, **<thsp>**, **<it>**, **<bf>**, **<rm>** и **<lang:XX>**; все остальные игнорируются.



```

line <тип> [ОПЦИИ]
  [ОПЦИИ]
  ...
  [ДААННЫЕ ЛИНИИ]
  ...
  [ОПЦИИ]
  ...
  [ДААННЫЕ ЛИНИИ]
  ...
endline

```

*Контекст:*

scrap

*Аргументы:*

- **<тип>** это зарезервированное слово, которое определяет тип линии. Поддерживаются следующие типы:

*символы хогов:* **wall** (стена), **contour** (контур), **slope** (склон)<sup>23</sup>, **floor-step** (уступ пола), **pit** (колодец), **ceiling-step** (уступ потолка), **chimney** (камин), **overhang** (нависание потолка), **ceiling-meander** (меандр, канал в потолке), **floor-meander** (меандр, канал в полу);

*заливка хогов:* **flowstone** (натечный каскад), **moonmilk** (мондмилх), **rock-border** (внешняя кромка глыбы)<sup>24</sup>, **rock-edge** (внутренняя кромка глыбы)<sup>25</sup>, **water-flow** (водоток);

*текстовые метки:* **label** (текстовая метка);

*специальные:* **border** (граница), **arrow** (стрелка), **section** (выносная линия сечения)<sup>26</sup>, **survey** (нитка хода)<sup>27</sup>, **map-connection** (линия выноса)<sup>28</sup>, **u**<sup>29</sup>.

*Опции командной строки:*

- **subtype <подтип>** ▸ определяет подтип линии. Для данных типов поддерживаются следующие подтипы:

*wall (стена):* **invisible** (не отображаемая), **bedrock** (коренная порода) (по умолчанию), **sand** (песчаная), **clay** (глинистая), **pebbles** (из крупной гальки), **debris** (из щебня), **blocks** (из блоков), **ice** (ледяная), **underlying** (лежащая ниже), **overlying** (лежащая выше), **unsurveyed** (не привязанная к опорным

5.4

<sup>23</sup> Линия склона обозначает верхнюю границу области склона. Необходимо указать **l-size** по меньшей мере для одной точки. Длина и ориентация уклона определяется указанными **l-size** и **orientation** (ориентация) в ближайших точках. Если ориентация отсутствует, то градиентные метки перпендикулярны линии наклона.

<sup>24</sup> Внешние границы крупных валунов. Если линия замкнута, она заполняется цветом фона.

<sup>25</sup> Внутренние края больших валунов.

<sup>26</sup> Линия показывает положение поперечного сечения. Если заданы обе контрольные точки (красные точки) кривой Безье (серая линия), то линия сечения (синяя) отображается до места перпендикуляра проекции (пунктир) первой и второй контрольных точек сечения. Кривая сечения не отображается.

5.3



<sup>27</sup> Нитка хода автоматически рисуется Therion'ом.

<sup>28</sup> Используется для указания соединения между картами (при смещении или же точками в разрез-развертке).

<sup>29</sup> Для определенных пользователем линейных символов.

точкам), **presumed** (предполагаемая), **pit** (стены колодца)<sup>30</sup>, **flowstone** (глыбовые), **moonmilk** (покрытые моңдмилхом);

**border** (граница): **visible** (видимая) (по умолчанию), **invisible** (не отображаемая), **temporary** (временная), **presumed** (предполагаемая);

**water-flow** (водоток): **permanent** (постоянный) (по умолчанию), **conjectural** (предполагаемый), **intermittent** (временный);

**survey** (нитка хода): **cave** (пещерная) (по умолчанию), **surface** (поверхностная) (по умолчанию, если нитка хода имеет флаг **surface**).

Подтип может быть указан также непосредственно в **<типе>** с использованием ':' в качестве разделителя<sup>31</sup>.

Любой подтип может использоваться с пользовательским типом (**u**). В этом случае вам также необходимо определить соответствующий метапост символ (смотрите главу *Новые символы карты*).

- **[ДАННЫЕ ЛИНИИ]** указывает либо координаты отрезка **<x> <y>**, либо координаты кривой Безье **<c1x> <c1y> <c2x> <c2y> <x> <y>**, где **c** указывает контрольную точку.
- **close <on/off/auto>** ▷ определяет, закольцована ли линия или нет.
- **mark <keyword>** ▷ используется для маркировки точки на линии (см. команду **join**).
- **orientation/orient <число>** ▷ ориентация символов на линии. Если не указано, она перпендикулярна линии с левой стороны.  $0 \leq \text{число} < 360$ .
- **outline <in/out/none>** ▷ определяет, служит ли линия границей скрапа. Значение по умолчанию – '**out**' для стен, '**none**' для всех других линий. Использовать **-outline in** для больших колонн и т.п.
- **reverse <on/off>** ▷ точки указаны в обратном порядке.
- **size <число>** ▷ ширина линии (размеры слева и справа установлены половине этого значения).
- **r-size <число>** ▷ размер линии справа.
- **l-size <число>** ▷ размер линии слева. Требуется для типа **slope** (склон).
- **smooth <on/off/auto>** ▷ сглаживание линии в данной точке. По умолчанию '**auto**'.
- **adjust <horizontal/vertical>** ▷ сдвигает точку линии, которая должна быть выровнена по горизонтали/вертикали с предыдущей точкой (или следующей точкой, если нет предыдущей точки). Результат – горизонтальный/вертикальный сегмент линии). Если у всех точек линии есть этот параметр, они выравниваются посередине координаты **y** или **x**, соответственно. Этот параметр не используется в проекции **plan**.
- **place <bottom/default/top>** ▷ порядок отображения изменений на карте.
- **clip <on/off>** ▷ указывает, подрезать ли символ границей скрапа.
- **visibility <on/off>** ▷ отображает/скрывает символ.
- **context <point/line/area> <тип символа>** ▷ (для использования с опциями **symbol-hide** (скрыть символ) и **symbol-show** (отображать символ) команды **layout**) будет скрыт/показан в соответствии с правилами для указанного **<типа символа>**.

<sup>30</sup> Обычно открытые с поверхности.

<sup>31</sup> Например, **border:invisible**.

### Специальные параметры:

- **altitude** <значение> ▷ могут быть указаны только с типом **wall** (стена). Этот параметр создает метку высоты на стене. Все высоты экспортируются как разница от Z сетки (по умолчанию 0). Если значение указано, оно дает разность высот точки на стене относительно ближайшей станции. Значение может иметь префикс с ключевым словом **"fix"**, тогда никакая ближайшая станция не будет принята во внимание; вместо этого используется абсолютное заданное значение. За значением можно указать единицы измерения. Например: **+4**, **[+4 m]**, **[fix 1510 m]**.
- **border** <on/off> ▷ эта опция может быть указана только с типом символа **'slope'** (склон). Он включает/выключает границу линии наклона.
- **direction** <begin/end/both/none/point> ▷ может использоваться только с типом **'section'** (сечение). Он указывает, куда следует поместить стрелку направления на линию сечения. По умолчанию **'none'**.
- **gradient** <none/center/point> ▷ может использоваться только с типом **'contour'** (контур) и указывает, где расположить градиентную метку на контурной линии. Если спецификация градиента отсутствует, то поведение зависит от набора символов (например, без галочки в UIS, галочка посередине в SKBB).
- **head** <begin/end/both/none> ▷ может использоваться только с типом **'arrow'** (стрелка) и указывает, куда поставить стрелу. По умолчанию **'end'** (в конце линии).
- **text** <строка> ▷ действует только для линий надписей.
- **height** <значение> ▷ высота **pit** (колодца) или **wall:pit** (входного колодца); 5.4  
доступная в METAPOST как числовая переменная **ATTR\_\_height**.

### Опции:

- **id** <идентификатор> ▷ идентификатор символа.

## 'area'

### Описание:

Area (область) определяется окружающими граничными линиями. Они могут быть любого типа, но должны быть перечислены друг за другом, и каждая пара последовательных линий должны пересекаться. Чтобы убедиться, что линии пересекаются даже после преобразования скрапа, вы можете продолжить границу на 1 см за стену хода – эти выступы будут автоматически обрезаны по границе скрапа. Вы можете использовать невидимую (**invisible**) границу для расположения внутри хода.

### Синтаксис:

```
area <тип>
  place <bottom/default/top>
  clip <on/off>
  visibility <on/off>
  ... ссылки на границы ...
endarea
```

### Контекст:

scrap

Аргументы:

- **<тип>** является одним из следующих: **water** (вода), **sump** (сифон), **sand** (песок), **debris** (щебень), **blocks** (навал глыб), **flowstone** (натечный каскад), **moonmilk** (мондмилх), **snow** (снег), **ice** (лед), **clay** (глина), **pebbles** (галька), **bedrock** (коренная порода)<sup>32</sup>, **u**<sup>33</sup>.

Опции командной строки:

- строки данных состоят из ссылок на линии границы (ID).
- **place <bottom/default/top>** ▸ изменяет порядок отображения на карте.
- **clip <on/off>** ▸ указывает, подрезать ли символ границей скрапа.
- **visibility <on/off>** ▸ отображает/скрывает символ.
- **context <point/line/area> <тип символа>** ▸ (для использования с опциями **symbol-hide** (скрыть символ) и **symbol-show** (отображать символ) команды **layout**) будет скрыт/показан в соответствии с правилами для указанного **<типа символа>**.

Опции:

- **id <идентификатор>** ▸ идентификатор символа.

## 'join'

Описание:

Соединение работает в двух режимах: оно объединяет либо два скрапа, либо две или несколько точек или линий на карте вместе.

При объединении более двух точек или линий используйте одну команду 'join' для всех из них, а не последовательность команд<sup>34</sup>.

При соединении скрапов соединяются только ходы. Лучше всего размещать соединения скрапов в ходе, которое будет как можно более простым, иначе вам нужно будет указать соединение для каждой пары объектов, которые должны быть соединены<sup>35</sup>.

Синтаксис:

**join <точка1> <точка2> ... <точкаN> [ОПЦИИ]**

Контекст:

none, scrap, survey

Аргументы:

- **<точкаX>** может быть идентификатором точечного символа или символа линии, необязательно сопровождаемый меткой символа линии **<идентификатор>:<метка>** (например, **podangl\_l31@podangl:mark1**). **<метка>** может быть также 'end' (конец линии) или индекс точки линии (где 0 это первая точка).

Частным случаем **<точка1>** и **<точка2>** могут быть идентификаторами скрапов, более близкие концы скрапов соединяются вместе.

---

<sup>32</sup> Пустая область, которая может быть использована для очистки фона.

<sup>33</sup> Для пользовательских символов области, может иметь произвольный подтип.

<sup>34</sup> Например, использовать **join a b c**, а не **join a b**, а затем **join b c**.

<sup>35</sup> Если вам нужен какой-либо объект, который отсекается границей скрапа, чтобы продолжить его на соседний скрап, используйте опцию **-clip off** для этого объекта.

Опции:

- **smooth** <on/off> ▷ указывает, должны ли две линии соединяться плавно.
- **count** <N> (при использовании со скрапами) ▷ Therion попытается подсоединиться к скрапам, которые соединяются в N месте/хода.

## 'equate'

Описание:

Устанавливает эквивалентность станций съемки.

Синтаксис:

**equate** <список станций>

Контекст:

none, survey

## 'map'

Описание:

Карта представляет собой набор либо скрапов, либо других карт одного и того же типа проекции. На карте можно включить съемку, тогда на карте будет отображаться нитка хода. Объект карты упрощает управление данными при выборе данных для вывода. Смотрите главу *Как собрать карту* для более подробного объяснения.

Синтаксис:

```
map <идентификатор> [ОПЦИИ]
... скрап, съемка или другие карты ...
break
... скрап, съемка или другие карты (следующий уровень)...
preview <above/below> <other map id>
endmap
```

Контекст:

none, survey

Аргументы:

- <идентификатор> ▷ идентификатор скрапа.

Опции командной строки:

- строки данных состоят из названий скрапа или карты. Обратите внимание, что вы не можете указывать их вместе.
- если вы указываете название карты, то вы можете указать смещение, в месте которого эта подкарта будет отображаться. Синтаксис следующий:  
<название карты> [<смещение X> <смещение Y> <единицы>]  
<above/below/none>.
- скрапы после **break** будут размещены на другом уровне.
- **preview** <above/below> <идентификатор другой карты> поместит контур другой карты в указанную позицию относительно текущей карты.

Карта отображается, только если она находится на уровне **map-level**, заданный командой **select**.

Используйте команду `revise`, если вы хотите добавить карты с более высоких уровней для отображения.

- `colo[u]r <цвет>` ▸ установить цвет карты; эта опция отменяет автоматический выбор, когда макет определяет `colour map-fg [map]`.

Опции:

- `projection/proj <plan/elevation/extended/none>` ▸ требуется, если карта содержит съемки.
- `title <строка>` ▸ описание объекта.
- 5.4 • `survey <идентификатор>` ▸ связывает съемку с картой (например, все статистические данные из этой съемки будут использоваться, когда эта карта будет выбрана для вывода).

## 'surface'

Описание:

Спецификация поверхности (рельефа). Ее можно отобразить двумя способами: в качестве отсканированной топографической карты (как в 2D-карте, так и в 3D-модели<sup>36</sup>) или поверхностная сетка – цифровая модель рельефа (только в 3D-модели).

Синтаксис:

```
surface [<имя>]
  cs <система координат>
  bitmap <имя файла> <калибровка>
  grid-units <единицы>
  grid <origin x> <origin y> <x spacing> <y spacing> <x count> <y count>
  grid-flip (none)/vertical/horizontal
  [grid data]
endsurface
```

Контекст:

none, survey

Опции командной строки:

- `cs <система координат>` ▸ система координат для калибровки растровой карты и определения начала сетки.
- `bitmap <имя файла> <калибровка>` ▸ сканированная топографическая карта.

Калибровка может иметь две формы:

1. `[X1 Y1 x1 y1 X2 Y2 x2 y2 [единицы]]`, где верхние регистры X/Y являются координатами изображения (пиксели, нижний левый угол равен 0 0), нижние регистры x/y – реальные координаты. Необязательный параметр единицы применяется к реальным координатам (метры по умолчанию).
  2. `[X1 Y1 станция1 X2 Y2 станция2]`, где верхние регистры X/Y представляют собой координаты изображения, а `станция1` и `станция2` являются именами станций съемки.
- `grid-units <единицы>` ▸ единицы, в которых задана сетка. Метры по умолчанию.
  - `grid <origin x> <origin y> <x spacing> <y spacing> <x count> <y count>`

<sup>36</sup> Вам нужно ввести данные о высоте, чтобы отобразить топографическую карту в 3D-модели. В настоящее время поддерживаются только карты JPEG в 3D.

`<origin x> <origin y>` ▷ указывают координаты нижнего левого (Ю-З) угла сетки.

`<x spacing> <y spacing>` ▷ расстояние между узлами сетки в направлениях З-В и Ю-С.

`<x count> <y count>` ▷ количество узлов в строке и количество строк, которые образуют сетку (см. ниже).

- `[grid data]` ▷ поток чисел, дающий высоту a.s.l. в узлах сетки. Начинается с начала сетки и заполняет сетку строками (в строке от З до В, строки от Ю до С).
- `grid-flip (none)/vertical/horizontal` ▷ полезно, если ваша сетка (экспортированная из другой программы) должна быть перевернута.

## **'import'**

*Описание:*

Считывает данные съемки в разных форматах (в настоящее время обрабатывается нитка хода в форматах \*.3d, \*.plt и \*.xyz). На станции съемки можно ссылаться в скрабах и т.д. При импорте 3D-файла SurveX'a станции вставляются в иерархии съемки, если в 3D-файле существует идентичная иерархия.

*Синтаксис:*

`import <имя файла> [ОПЦИИ]`

*Контекст:*

survey / all<sup>37</sup>

*Опции:*

- `filter <префикс>` ▷ если указано, то будут импортированы только станции с заданным префиксом и замеры между ними. Префикс будет удален из имен станций.
- `surveys (create)/use/ignore` ▷ указывает, как импортировать структуру съемки (работает только с файлами .3d).
  - `create` ▷ сплит-станции в подсъемки, если подсъемок не существуют, то создаст их.
  - `use` ▷ разделенные станции в существующие подсъемки.
  - `ignore` ▷ не разбивать станции на подсъемки.
- `cs <система координат>` ▷ система координат для станций с фиксированными координатами.
- `calibrate [<x> <y> <z> <X> <Y> <Z>]` ▷ координаты в импортированном файле смещаются от координат в нижнем регистре к координатам верхнего регистра.

---

<sup>37</sup> Только с .3d-файлами, в которых задана структура съемки.



## 'grade'

*Описание:*

Эта команда используется для сохранения предопределенных точных данных нитки хода. Смотрите опцию `sd` в описании команды `centreline`.

*Синтаксис:* `grade <идентификатор>`

```
...  
[<quantity list> <значение> <единицы>]  
...  
endgrade
```

*Контекст:*

all

## 'revise'

*Описание:*

Эта команда используется для установки или изменения свойств уже существующего объекта.

*Синтаксис:*

Синтаксис этой команды для объекта, созданного с помощью команды "single line":

`revise идентификатор [-опция1 значение1 -опция2 значение2 ...]`

Для объектов, созданных с помощью команд "multi line" используется синтаксис:

`revise идентификатор [-опция1 значение1 -опция2 значение2 ...]`

```
...  
опцияX значениеX  
данные  
...  
endrevise
```

*Контекст:*

all

*Аргументы:*

В качестве идентификатора всегда должен указываться идентификатор объекта, который вы хотите изменить.

## Пользовательские атрибуты

Объекты *survey*, *centreline*, *scrap*, *point*, *line*, *area*, *map* и *surface* могут содержать пользовательские атрибуты в форме `-attr <имя> <значение>`. `<Имя>` может содержать буквенно-цифровые символы, `<значение>` – строку.

Пользовательские атрибуты используются при экспорте карты в зависимости от формата вывода:

- при экспорте в *shapefile* они записываются непосредственно в файл dbf;
- в картах, сгенерированных с использованием METAPOST (PDF, SVG) атрибуты записываются в исходный файл METAPOST как строки



(именуемые `ATTR_<имя>`) и могут быть оценены и использованы пользователем в макросах определения символов.

Вы можете проверить наличие такой переменной, используя: `if known ATTR_<имя>: ... fi.`

## XTherion

XTherion – графический пользовательский интерфейс для Therion'a. Он помогает в создании файлов входных данных. В настоящее время он работает в трех основных режимах: текстовый редактор, редактор карт и компилятор<sup>38</sup>.

Его не обязательно использовать для Therion'a – вы можете редактировать входные файлы в своем любимом текстовом редакторе и запускать Therion из командной строки. XTherion также не является единственным графическим интерфейсом, который можно использовать с Therion'ом. Можно написать лучшую, более удобную для пользователя, более WYSIWYG-ориентированную, быструю, более надежную и удобную в использовании. Есть желающие?

В этом руководстве не описываются такие знакомые вещи, как 'если вы хотите сохранить файл, перейдите в меню Файл и выберите Сохранить или нажмите Ctrl-s'. Просмотрите верхнее меню, чтобы почувствовать XTherion.

Для каждого режима работы есть дополнительное меню справа или слева. Подменю могут быть свернуты; вы можете развернуть их, нажав кнопку меню. Для большинства меню и кнопок в строке состояния есть короткое описание, поэтому нетрудно догадаться о значении каждого из них. Показ подменю сбоку может быть настроен пользователем. Щелкните правой кнопкой мыши по кнопке меню и выберите в меню тот пункт, который вы хотите здесь отображать.

### XTherion – текстовый редактор

Текстовый редактор XTherion предлагает некоторые интересные функции, которые могут помочь в создании текстовых входных файлов: поддержка кодировки Unicode и возможность открытия нескольких файлов<sup>39</sup>.

Чтобы упростить ввод данных, он поддерживает форматирование таблиц нитки хода. Для ввода данных существует меню *Таблица данных*. Она может быть настроена на ввод данных пользователя, нажав кнопку *Определить формат данных*, когда курсор находится под спецификацией данных (опция 'дата' в команде 'centreline').

### XTherion – редактор карт

Редактор карт позволяет вам рисовать и редактировать карту полностью в интерактивном режиме. Но не ожидайте слишком многого. XTherion не является редактором WYSIWYG. Он отображает только позицию, а не фактическую форму, нарисованных точек или линий. Визуально нет никакой разницы между геликтитом и текстовой меткой – оба они

---

<sup>38</sup> Здесь мы обсуждаем создание данных, поэтому в этом разделе описаны только два первых режима. Функции компилятора смотрите в главе Обработка данных.

<sup>39</sup> Кодировка файла указана в первой строке файла. Эта строка скрыта XTherion'ом и может быть доступна только косвенно, используя правое меню.

отображаются как простые точки. Тип и другие атрибуты любого объекта указываются только в меню *Точка* и *Линия*.

*Упражнение:* Найдите две существенные причины, почему карта, нарисованная в XTherion, не может быть идентична выходному файлу Therion'a. (Если вы ответите на это, вы узнаете, почему XTherion никогда не будет истинным редактором WYSIWYG. Лень авторов – не правильный ответ.)

Начнем с описания типичного использования редактора карт. Во-первых, вам нужно решить, какую часть пещеры (какой скрап) вы рисуете<sup>40</sup>.



После создания нового файла в редакторе карт вы можете загрузить одно или несколько **изображений** (сканированные эскизы съемки пещеры<sup>41</sup>) в качестве подложки для рисования. Нажмите кнопку *Вставка* в меню *Фоновые изображения*. К сожалению, из-за ограничений языка Tcl/Tk, поддерживаются только изображения в форматах GIF, PNM и PPM (плюс PNG и JPEG, если вы установили расширение tkImg). Кроме того, XTherion поддерживает XVI (XTherion vector image), в котором отображается нитка хода и LRUD, и данные PocketToro экспортируются в формат Therion'a (см. ниже). Все добавленные изображения помещаются в верхний левый угол рабочей области. Переместить их можно двойным щелчком правой кнопкой мыши на изображении и перетаскиванием или через меню. Для повышения производительности на более медленных компьютерах можно временно выгрузить неиспользуемое изображение из памяти, сняв флажок *показать*. Можно открыть существующий файл без загрузки фоновых изображений с помощью меню *Открыть (без картинок)*<sup>42</sup>.

Размер и масштабирование **области рисования** настраивается в соответствующем меню. *Авто* вычисляет оптимальный размер рабочей области в соответствии с размерами и позициями загруженных фоновых изображений.

После этих этапов подготовки вы готовы к рисованию или, точнее, для **создания файла данных карты**. Важно помнить, что вы на самом деле создаете текстовый файл, который должен соответствовать синтаксису, описанному в главе *Формат данных*. На самом деле в редакторе карт используются только несколько команд Therion'a: многострочная команда *scrap ... endscrap* может содержать команды *point*, *line* и *area* (смотрите главу *Формат данных*). Это соответствует этапу ручного рисования карты, которая строится из точек, линий и заполненных областей.

Итак, первым шагом является определение **скрапа** с помощью *scrap ... endscrap* многострочной команды. В меню *Команды в файле* выберите подменю *Действие* и выберите *Вставить скрап*. Это изменит кнопку *Действие* на *Вставить скрап*, если у нее было другое значение. После нажатия этой кнопки в начало файла будет вставлен новый скрап. Вы должны видеть строки

<sup>40</sup> В одном файле можно нарисовать несколько скрапов, в этом случае все неактивные скрапы отображаются желтым.

<sup>41</sup> XTherion не может масштабировать и поворачивать отдельные изображения, поэтому используйте ту же ориентацию, масштаб и DPI для всех изображений, используемых в одном и том же скрапе.

<sup>42</sup> Примечание: Therion никак не использует фоновые изображения, если вы не назначили их для определенного скрапа с помощью опции *-sketch*.

scrap - scrap1  
endscrap  
end of file

в окне предварительного просмотра над кнопкой *Вставить скрап*. Это окно представляет собой упрощенный вывод текстового файла, который будет сохранен XTherion'ом. Показываются только команды (*scrap*, *point*, *line*, *text* – почему так, смотрите ниже) и их типы (для *point* и *line*) или ID (для *scrap*).

Полное содержимое любой команды отображается в меню *Просмотр команды*.

Для изменения ранее созданных команд есть дополнительные меню – например *Скрап* для команды *scrap*. Здесь вы можете изменить ID (очень важно!) и другие опции. Подробнее смотрите главу *Формат данных*.

Теперь можно вставить некоторые **точечные символы**. Как и в случае вставки скрапа, перейдите в меню *Команды в файле*, нажмите подменю *Действие* и выберите *Вставить точку*; затем нажать кнопку с изменившимся названием на *Вставить точку*. Сочетание клавиш для этого – Ctrl-p. Затем нажмите на нужное место в рабочей области, и вы увидите синюю точку, представляющую символ точки. Ее атрибуты можно настроить в меню *Точка*. Вы останетесь в режиме 'вставки' – каждый щелчок по рабочей области добавляет новый символ точки. Старайтесь не нажимать дважды в одном месте – тогда вы вставите два точечных символа в одном и том же месте! Чтобы выйти из режима 'вставить', нажмите клавишу *Esc* на клавиатуре или кнопку *Выбрать* в меню *Команды в файле*.

Каков порядок команд в выходном файле? Точно такой же, как в меню *Команды в файле*. Вновь созданные точечные, линейные и текстовые объекты добавляются перед текущей выделенной строкой. Можно изменить порядок, выбрав строку и нажав кнопки *Вниз*, *Вверх* или *Переместить* в меню *Команды в файле*. Таким образом вы также можете перемещать объекты между скрапами.

**Рисование линий** аналогично рисованию в других программах редактирования векторной графики, которые работают с кривыми Безье. (Угадайте, как войти в режим вставки линии кроме использования сочетания Ctrl-l.) Нажмите, где должна быть первая точка, затем перетащите мышь с нажатой левой кнопкой и отпустите ее в том месте, где должна быть первая контрольная точка. Затем нажмите где-нибудь еще (эта точка будет второй точкой кривой) и перетащите мышь (отрегулируйте вторую контрольную точку предыдущей дуги и первую контрольную точку следующей дуги, одновременно). Если это объяснение кажется слишком неясным, вы можете поработать в некоторых стандартных векторных редакторах. Линия будет завершена после выхода из режима вставки. Начало и ориентация линии отмечены небольшой оранжевой галочкой в первой точке.

Для символов линии существуют два управляющих меню: *Линия* и *Точка линии*. Сначала устанавливаются атрибуты для всей кривой, такие как тип или имя. Важным является чек-бокс *обратная*: Therion требует ориентированных кривых, и нет ничего необычного в том, что вы начинаете рисовать с неправильного конца. Меню *Точка линии* позволяет вам отрегулировать атрибуты любой выбранной точки на линии, например, сглаживание кривой в этой точке (которая включена по умолчанию) или наличие соседних контрольных точек ('«' и '»').

**Области** определяются окружающими их линиями. Нажмите *Вставить область*, а затем щелкните строки окружающие нужную область. Они автоматически вставляются в *Область* и называются (если они еще не

названы). Альтернативный способ вставить их как **text**<sup>43</sup>, содержимое которого (введенное в меню *Редактор* в редакторе карты) обычно многострочная команда **area ... endarea** (смотрите раздел *Формат данных*.)

Если вы нарисуете несколько скрапов с **none** проекцией, необходимо **откалибровать** область рисования. Масштаб можно определить только одним способом в XTherion'e – используя координаты двух точек (заданных как в системе координат изображения, так и в 'реальной' системе координат).

После выбора скрапа (щелкните по его заголовку в меню *Команды в файле*) появятся два небольших красных квадрата, соединенных красной стрелкой (по умолчанию они будут в нижних углах области рисования). Вы должны перетащить их в точки с известными координатами – обычно пересечения линий сетки на миллиметровой на отсканированном чертеже. Если вы не видите их, вы можете:

- нажать кнопку *Масштаб* в меню *Скрап* и щелкнуть два разных места на изображении, где должны быть конечные точки калибровочной стрелки, или
- переместить указатель мыши в нужную позицию, прочесть координаты указателя и ввести эти координаты в *масштабирующие точки для картинки* в меню *Скрап*. После заполнения пар координат X1, Y1 и X2, Y2 стрелка калибровки будет перемещаться соответственно.

Затем вам нужно ввести реальные координаты этих точек (X1, Y1, X2, Y2).

В **режиме выбора** вы можете выбрать существующие линейные или точечные объекты и установить их атрибуты в соответствующих меню, переместить их или удалить их (Ctrl-d или *Кнопка действия* в меню *Команды в файле* после установки *Действие* на *Удалить*).

Существует меню *Поиск и выделение*, которое позволяет легко переключаться между объектами и показывать элементы, которые вы не видите при взгляде на изображение. Например, если вы введете слово 'station' и нажмете *Показать все*, все станции на экране станут красными.

XTherion не выполняет проверку синтаксиса; он только записывает объекты с атрибутами в текстовый файл. Любые ошибки обнаруживаются только при обработке этих файлов с помощью Therion'a.

**СОВЕТ.** Ввод символов одного и того же типа одновременно экономит вам много времени, потому что вам не нужно менять тип символа и параметры заполнения для каждого нового символа. Поле *Опции* сохраняет старое значение, и достаточно изменить всего несколько символов<sup>44</sup>. Рекомендуется начать с рисования всех станций съемки (не забудьте дать им имена (номера) в соответствии с настоящими именами в команде *centreline*), далее все хода, за которыми следуют все остальные точечные символы, линии и области. В конце рисуем поперечные сечения.

---

<sup>43</sup> ВНИМАНИЕ! Команда **text** – это не команда Therion'a! Это всего лишь псевдоним для блока произвольного текста в XTherion'e. В файле, сохраненном XTherion'ом, будет только то, что вы введете в Редакторе или смотрите в Просмотре команд. Это может быть определение области или что угодно, например комментарий, начинающийся с символа '#'.

<sup>44</sup> В случае станций съемки XTherion автоматически увеличивает номер станции для следующего вставленного символа.

## Дополнительные инструменты

5.3 **Помощь/Привязать изображение** создает файл MAP совместимый с OziExplorer на основе геоданных включенных в карту PDF<sup>45</sup>.

Если карта в формате PDF была преобразована в растровое изображение с использованием внешней программы, конвертер использует растровое изображение и pdf-карту с тем же базовым именем, расположенным в том же каталоге, для вычисления калибровочных данных.

Если используется непосредственно файл PDF, то вам необходимо установить DPI и формат вывода перед автоматическим преобразованием<sup>46</sup> в растровый формат.

**Данные PocketToro** экспортированные в формате Therion'a<sup>47</sup> из приложения PocketToro можно импортировать в текстовом редакторе, а также в редакторе карт (*Файл → Импорт → PocketToro therion export* и *Фоновые изображения → Вставить → PocketToro therion export*). Тот же файл используется для обоих импортов. Импорт эскиза напрямую не создает данные скрапа. Рисунок просто отображается на фоне как сканированное растровое изображение, и должен быть оцифрован вручную.

5.3

## Сочетания клавиш и мыши в редакторе карт

### Общие

- Ctrl+Z ▷ отменить изменения;
- Ctrl+Y ▷ вернуть изменения;
- F9 ▷ компилировать текущий проект;
- для выбора объекта в списке с помощью клавиатуры: переключайтесь с помощью 'Tab' в желаемый список; перемещаться на нужный объект (объект подчеркивается); нажмите 'Пробел';
- PageUp/PageDown ▷ скролинг вверх/вниз в боковой панели;
- Shift+PageUp/PageDown ▷ скролинг вверх/вниз в окне Команды в файле.

### Область рисования и фоновые изображения

- клик правой кнопкой мыши ▷ скролинг области рисования;
- двойной щелчок правой кнопкой мыши по изображению ▷ перемещение изображения.

### Вставка скрапа

- Ctrl+R ▷ вставить скрап.

<sup>45</sup> Может присутствовать до девяти калибровочных точек в виде фиксированных станций в нитке хода с использованием геодезической системы координат.

<sup>46</sup> Ghostscript и convert должны быть установлены в вашей системе. Обратите внимание, что установщик Windows не содержит ghostscript.

<sup>47</sup> Это специальный текстовый формат, который нужно импортировать с помощью XTherion'a и не может обрабатываться непосредственно Therion'ом.



### *Вставка линии*

- Ctrl+L ▷ вставить новую линию и войти в режим 'вставка точек линии';
- щелчок левой кнопкой мыши ▷ вставить точку линии (без контрольных точек)
- Ctrl+щелчок левой кнопкой мыши ▷ вставить точку линии очень близко к существующей точке (обычно она вставляется на ближайшую существующую точку);
- щелчок левой кнопкой мыши+тянуть ▷ вставить точку линии (с контрольными точками);
- удерживать Ctrl при перемещении ▷ изменить расстояние предыдущей контрольной точки;
- щелчок левой кнопкой мыши+переместить контрольную точку ▷ переместить ее положение;
- щелчок правой кнопкой мыши на одной из предыдущих точек ▷ выбирает предыдущую точку в режиме вставки (полезно, если вы хотите изменить также направление предыдущей контрольной точки);
- Esc или щелчок левой кнопкой мыши на последней точке ▷ завершение вставки линии;
- щелчок левой кнопкой мыши на первой точке линии ▷ замкнуть линию и завершить вставку линии.

### *Редактирование линии*

- щелчок левой кнопкой мыши+перемещение ▷ перемещение точки линии;
- Ctrl+щелчок левой кнопкой мыши+перемещение ▷ перемещение точки линии близко к существующей точке (обычно она перемещается на ближайшую существующую точку);
- щелчок левой кнопкой мыши на контрольной точке+перемещение ▷ переместить контрольную точку.

### *Добавление точки линии*

- выберите точку, перед которой вы хотите вставить новые точки; вставить необходимые точки; нажмите Esc или щелкните левой кнопкой мыши по выбранной вами точке в начале.

### *Удаление точки линии*

- выберите точку, которую хотите удалить; нажмите *Править линию* → *Удалить точку* в панели *Линия*.

### *Разделение линии*

- выберите точку, в которой вы хотите разделить линию; нажмите *Править линию* → *Разделить линию* в панели *Линия*.

### *Вставка точек*

- Ctrl+P ▷ переход в режим 'вставки точки';
- щелчок левой кнопкой мыши ▷ вставка точки в заданной позиции;
- Ctrl+щелчок левой кнопкой мыши ▷ вставка точки очень близко к существующей точке (обычно она будет вставлена на ближайшую точку);
- Esc ▷ выход из режим 'вставки точки'.

### *Редактирование точки*

- щелчок левой кнопкой мыши+перетаскивание ▷ перемещение точки;
- Ctrl+щелчок левой кнопкой мыши+перетаскивание ▷ перемещение точки близко к существующей точке (обычно она перемещается на ближайшую существующую точку);
- щелчок левой кнопкой мыши+перетаскивание стрелок точки ▷ изменение ориентации и/или размера точки (согласно заданным переключателям в панели управления точки).

### *Вставка области*

- нажмите Ctrl+A или *Команды в файле* → *Вставить* → *область*, чтобы переключиться в режим 'вставки границ области';
- щелчок правой кнопкой мыши на линиях, которые окружают желаемую область;
- Esc, чтобы закончить ввод границ области.

### *Редактирование области*

- выберите область, которую вы хотите отредактировать;
- нажмите 'Вставить' в *Область*, чтобы вставить другие границы в текущую позицию курсора;
- нажмите 'Вставить по ID', чтобы вставить границу с заданным ID в текущую позицию курсора;
- нажмите 'Delete', чтобы удалить выбранную границу области.

### *Выбор существующего объекта*

- щелчок левой кнопкой мыши ▷ выбрать верхний объект;
- щелчок правой кнопкой мыши ▷ выбрать объект под верхним объектом (полезно, когда несколько точек лежат друг над другом).

## **Несколько мыслей о Therion'e**

Несмотря на то, что все (ну, почти все) о входных файлах Therion'a было сказано, в этой главе приведены некоторые дополнительные советы и подсказки.

### **Как ввести нитку хода?**

Основным блоком построения карт является команда [centreline](#). Если пещера больше нескольких метров, неплохо было бы разделить данные в большем количестве файлов и отделить данные нитки хода от данных карты.

Обычно мы используем один [\\*.th](#) файл содержащий нитку хода для каждой отдельной съемки. Удобно начинать с пустого файла шаблона, как показано ниже, где точки будут заменены соответствующим текстом.

```

encoding ISO8859-1
survey ... -title "..."
  centreline
    team "..."
    team "..."
    date ...
    units clino compass grad
    data normal from to compass clino length
    ... ..
  endcentreline
endsurvey

```

Чтобы создать уникальное пространство имен команда `centreline` заключена в команду `survey ... endsurvey`. Это полезно, если `survey` имеет то же имя, что и файл, который его содержит<sup>48</sup>. На точки можно будет ссылаться используя символ `@` – смотрите описание команды `survey`.

Для действительно больших пещер можно построить иерархическую структуру каталогов. В этом случае мы создаем один специальный файл с именем `INDEX.th` который включает все остальные `*.th` файлы из данного каталога и содержит команды `equate` для определения связей между съемками.

## Как рисовать карты?

Самое главное – придумать деление пещеры на скрапы. Скрап является основным блоком карты. Попытка подобрать скрап к соответствующему `*.th` файлу с ниткой хода одной съемки – это почти всегда плохая идея. Причина в том, что соединения между скрапами должны быть как можно более простыми. Скрапы в целом независимы от иерархии нитки хода, поэтому старайтесь не привязываться к съемке когда рисуете карты и выбираете лучшие объединения скрапов.

Мы обычно вставляем карты в предпоследний уровень в иерархии съемок<sup>49</sup>. Каждый скрап может содержать произвольную часть любой съемки на последнем уровне иерархии. Например, есть `survey main` который содержит `surveys a, b, c` и `d`. `Surveys a – d` содержат данные нитки хода от четырех съемок, и каждый из них находится в отдельном файле. Есть карта `main_map` которая содержит скрапы `s1` и `s2`. Если `main_map` находится в `survey main`, скрап `s1` может содержать часть нитки хода из `survey a`, полную нитку хода `b` и часть `c`; `s2` будет содержать часть `a` и `c` и полную нитку `d`. Названия станций `survey` будут иметь символ `@` (например, `1@a`)<sup>50</sup>.

Скрапы обычно хранятся в `*.th2` файлах. Каждый файл может содержать несколько скрапов. Чтобы данные были хорошо организованы, мы приняли несколько соглашений об именах файлов: в файле `foo.th2` все скрапы именуются `foo_si`, где `i – 1, 2` и так далее. Поперечные сечения именуются `foo_ci`, линии `foo_li` и т.п. Это очень помогает с большими пещерными системами; если упоминается какой-то скрап, вы сразу же знаете в каком файле он был определен.

<sup>48</sup> Например `survey entrance` в файле `entry.th`.

<sup>49</sup> Помните, что съемки создают пространства имен, поэтому вы можете ссылаться только на объекты в данной съемке и на все подсъемки.

<sup>50</sup> Если вы подключаете файлы карт в начале `survey`, вы можете ссылаться на любую станцию в любом скрапе, что представляет собой очень гибким инструмент. С другой стороны вы можете использовать более длинные имена в ссылках на станции, например `3@dno.katakomby.jmn.dumbier`.



Можно создавать один файл `INDEX.th2` на каталог, который подключает все `*.th2` файлы, описывает maps и объединения скрапов.

При рисовании скрапов вы должны проверить, правильно ли определен контур: все линии создающие внешнюю границу ходов должны иметь опцию `-outline out`; все линии окружающие внутренние колонны опцию `-outline in`. Границы скрапа не могут пересекаться, иначе внутренняя сторона скрапа не может быть определена. Есть два простых теста, которые определяют правильность скрапа:

- не выводится предупреждение METAPOST `"scrap outline intersects itself"`;
- когда вы устанавливаете цвет заливки ходов (`color map-fg <номер>` опция в `layout`), вы можете видеть что Therion считает внутренней областью скрапа.

## Как создавать модели?

Модель создается из внешних линий скрапов. Высота и глубина хода вычисляются по символьным точкам `высота хода` (`passage-height`) и `размер хода` (`dimensions`).

## В глубь Therion'a

### Как карта собирается в единое целое

В этой главе объясняется работа опций `-clip`, `-place`, `-visibility` и `-context` команд `point`, `line` и `area`. Также объясняются опции `color`, `transparent`, `symbol-hide` и `symbol-show` команды `layout`.

При экспорте карты Therion должен определить три атрибута для каждого из символов – точки, линии или области: видимость, подрезание и упорядочение.

(1) Символ отображается, если верно следующее:

- он имеет опцию `-visibility` (видимый) со значением `on` (все символы по умолчанию);
- он не был скрыт опцией `-symbol-hide` (скрыть символ) в `layout`;
- если его опция `-context` (контекст) установлена, а соответствующий символ не был скрыт опцией `-symbol-hide` (скрыть символ) в `layout`.

Экспортируются только видимые символы.

(2) Некоторые символы могут подрезаться абрисом скрапа. По умолчанию это все следующие объекты:

- *точечные символы*: символы наполнения ходов (коренная порода... гуано);
- *линейные символы*: все символы линий, которые не имеют опцию `-outline`, за исключением `section` (выносная линия сечения), `arrow` (стрелка), `label` (текстовая метка), `gradient` (уклон хода) и `water-flow` (водоток);
- *символы области*: все.

Значение по умолчанию может быть изменено с помощью опции `-clip`, если это разрешено для определенного символа. Все остальные символы не подрезаются границей скрапа.

Упорядочивание: каждый символ относится к одной из следующих групп, которые последовательно выводятся:

- `bottom` ▷ все символы с опцией `-place bottom`;
- `default-bottom` ▷ все символы **области** по умолчанию;
- `default` ▷ символы, которые не принадлежат ни к какой другой группе;
- `default-top` ▷ `ceiling-step` (уступ потолка) и `chimney` (камин) по умолчанию;
- `top` ▷ все символы с опцией `-place top`.

Порядок символов внутри каждой группы соответствует порядку команд во входном файле<sup>51</sup>: символы, которые идут первыми, нарисованы последними (т.е. они отображаются в верхней части каждой группы).

Теперь мы готовы описать как строится карта (или атлас):

- область карты заполняется `color map-bg`;
- растровые изображения поверхности отображаются, если `surface` установлен в `bottom`;
- Для каждого скрапа: абрисы заполняются белым;
- рисуется сетка если `grid` установлен в `bottom`;
- анонс внизу<sup>52</sup> заполняется `color preview-below`;
- Для каждого уровня<sup>53</sup>:

    НАЧАЛО отсечения

        Для каждого скрапа: абрис заполняется `color map-fg`

        Для каждого скрапа: символы **области** заполняются и отсекаются до границы скрапа

    КОНЕЦ отсечения

НАЧАЛО отсечения текстовых меток (для всех текстовых меток на этом и верхних уровнях)

    Для каждого скрапа:

        рисуются все символы прошедшие отсечение (за исключением `line survey`), упорядочиваются снизу вверх

        рисуются `line survey` символы

        отсекаются границы скрапов

    Для каждого скрапа:

        рисуются все не отсеченные символы (за исключением `point station` и всех текстовых меток), упорядочиваются

---

<sup>51</sup> Или в меню Команды в файле XTherion'a.

<sup>52</sup> Согласно спецификации опции `preview` в команде `map`.

<sup>53</sup> Уровень – это коллекция скрапов не разделенная `break` в команде `map`.

снизу вверх

рисуются [point station](#) символы

КОНЕЦ отсечения текстовых меток

ДЛЯ каждого скрапа: рисуются все (точечные и линейные) текстовые метки (в том числе [wall-altitude](#));

- анонс рисуется поверх с [color preview-above](#);
- растровые изображения поверхности отображаются если [surface](#) установлен в [top](#);
- сетка отображается если [grid](#) установлен в [top](#).

Невозможно дважды войти в одну реку.  
Ποταμοῖς τοῖς αὐτοῖς ἐμβαίνομεν τε καὶ οὐκ ἐμβαίνομεν.  
—Heraclitus of Ephesus, 6th/5th century BC

## Обработка данных

Помимо файлов данных, содержащих данные съемки, Therion использует файл конфигурации, содержащий инструкции о том, как данные должны обрабатываться.

### Конфигурационный файл

Имя конфигурационного файла можно указать в качестве аргумента для Therion'a. По умолчанию Therion ищет файл с именем `thconfig` в текущем рабочем каталоге. Он обрабатывается как любой другой файл (т.е. одна команда на строку, пустые строки или строки, начинающиеся с '#', игнорируются; строки, заканчивающиеся на обратную косую черту, продолжаются в следующей строке.) Ниже список поддерживаемых в данный момент команд.

#### 'system'

Позволяет выполнять системные команды во время компиляции<sup>54</sup>. Обычно Therion ожидает завершения подпроцесса. Если вы хотите продолжить компиляцию без ожидания, используйте команду `<command> &` для Linux и `start <command>` для Windows.

#### 'encoding'

Работает аналогично команде `encoding` в файлах данных – задает кодировку символов.

#### 'language'

Синтаксис:

- `language <xx_[YY]>`

5.3 Устанавливает язык вывода для переводимых текстов.

#### 'cs'

Синтаксис:

- `cs <система координат>`

5.3 Вне команды `layout` определяет систему координат для вывода. Невозможно указать несколько систем координат для разных выходных данных (последнее указание `cs` используется для всех выходных файлов).

Если `cs` не определяется в файле конфигурации, то будет использован первый `cs` в файлах данных.

Внутри `layout` определяется система координат для последующих данных о местоположении (`origin`, `grid-origin`).

---

<sup>54</sup> Например, чтобы открыть или обновить внешний просмотрщик PDF.

## 'sketch-warp'

*Синтаксис:*

- `sketch-warp <алгоритм>`

Указывает, какой алгоритм преобразования скрапа (морфинг) использовать. Возможные алгоритмы: `line` – по умолчанию; `plaquette` – разработанный Marco Corvi.

## 'input'

Работает как команда `input` в файлах данных – подключает другие файлы.

## 'source'

*Описание:*

Указывает, какие файлы съемки (данных) Therion должен читать. Здесь вы можете указать несколько файлов; по одной в каждой строке. Вы также можете указать их с помощью опции `-s` командной строки (см. ниже).

Также можно ввести некоторые небольшие фрагменты кода непосредственно в файле конфигурации, используя многострочный синтаксис.

*Синтаксис:*

```
source <имя файла>
или
source
...команды Therion'a...
endsource
```

*Аргументы:*

- `<имя файла>`

## 'select'

*Описание:*

выбирает объекты (съемки и карты) для экспорта. По умолчанию выбираются все объекты съемки. Если карта не выбрана, то выбираются все скрапы принадлежащие выбранным съемкам.

Если в данных нет скрапов или карт, то на карте будет отображаться нитка хода из всех съемок.

При экспорте карт в разных проекциях вам нужно выбирать их для каждой проекции отдельно.

*Синтаксис:*

```
select <объект> [ОПЦИИ]
```

*Аргументы:*

- `<объект>` ▷ любая съемка или карта, имеющие идентификатор ID.

*Опции:*

- **recursive** <on/off> ▷ действует только при выборе съемки. По умолчанию установлен в **on**, все подсъемки данной съемки рекурсивно выбираются/не выбираются.
- **map-level** <число> ▷ действует только при выборе карты. Определяет уровень, на котором прекращается расширение карты для экспорта в виде атласа. По умолчанию используется 0; если указано 'basic', расширение выполняется до основных карт (т.е. все). *Примечание:* Предварительный просмотр карт отображается только в соответствии с картами в текущем **map-level**.
- **chapter-level** <число> ▷ действует только при выборе карты. Определяет уровень, при котором прекращается расширение главы для экспорта в виде атласа. По умолчанию используется 0; если используется '-' или '.', то для этой карты не экспортируется ни одна глава. Если опция **title-pages** в **layout** включена, то каждая глава начинается с титульной страницы.

## **'unselect'**

*Описание:*

Отменяет выбор объектов для экспорта.

*Синтаксис:*

**unselect** <объект> [ОПЦИИ]

*Аргументы:*

Так же, как и в команде **select**.

*Опции:*

Так же, как и в команде **select**.

## **'text'**

*Описание:*

Задаёт перевод текста по умолчанию для вывода.

*Синтаксис:*

**text** <ID языка> <therion текст> <перевод>

*Аргументы:*

- <ID языка> ▷ стандартный идентификатор языка ISO (например, **en** или **en\_GB**).
- <therion текст> ▷ текст Therion'а для перевода. Список текстов и доступных переводов смотрите в файле **thlang/texts.txt**;

## **'layout'**

*Описание:*

Задаёт макет для 2D-карт. Настройки, применяемые к режиму атласа, отмечены 'A'; к режиму карт – 'M'.

*Синтаксис:*

layout <идентификатор> [ОПЦИИ]  
 сору <идентификатор исходного макета>  
 cs <система координат>  
 north <true/grid>  
 scale <длина изображения> <реальная длина>  
 base-scale <длина изображения> <реальная длина>  
 units <metric/imperial>  
 rotate <число>  
 symbol-set <набор символов>  
 symbol-assign <point/line/area/group/special> <тип символа>  
     <набор символов>  
 symbol-hide <point/line/area/group/special> <тип символа>  
 symbol-show <point/line/area/group/special> <тип символа>  
 symbol-colour <point/line/area/group/special> <тип символа> <цвет>  
 min-symbol-scale <масштаб>  
 fonts-setup <tinysize> <smallsize> <normalsize> <largesize> <hugesize>  
 size <ширина> <высота> <единицы>  
 overlap <значение> <единицы>  
 page-setup <размеры> <единицы>  
 page-numbers <on/off>  
 exclude-pages <on/off> <список>  
 title-pages <on/off>  
 nav-factor <фактор>  
 nav-size <х-размер> <у-размер>  
 transparency <on/off>  
 opacity <значение>  
 surface <top/bottom/off>  
 surface-opacity <значение>  
 sketches <on/off>  
 layers <on/off>  
 grid <off/top/bottom>  
 grid-origin <x> <y> <z> <единицы>  
 grid-size <x> <y> <z> <единицы>  
 grid-coords <off/border/all>  
 origin <x> <y> <z> <единицы>  
 origin-label <х-метка> <у-метка>  
 own-pages <число>  
 page-grid <on/off>  
 legend <on/off/all>  
 legend-columns <число>  
 legend-width <n> <единицы>  
 map-comment <текст>  
 map-header <x> <y> <off/n/s/e/w/ne/nw/se/sw/center>  
 map-header-bg <on/off>  
 map-image <x> <y> <n/s/e/w/ne/nw/se/sw/center> <имя файла>  
 statistics <explo/topo/carto/copyright all/off/number>  
     <explo/topo-length on/off>  
 scale-bar <длина> <единицы>  
 survey-level <N/all>  
 language <xx[\_YY]>  
 colour/color <элемент> <цвет>  
 debug <on/all/first/second/scrap-names/station-names/off>  
 doc-author <текст>  
 doc-keywords <текст>  
 doc-subject <текст>  
 doc-title <текст>



```
code <metapost/tex-map/tex-atlas>
endcode
endlayout
```

Аргументы:

`<идентификатор>` ▷ идентификатор макета (используется в команде `export`).

Опции командной строки:

- `copy <идентификатор исходного макета>` ▷ устанавливаются свойства макета, которые берутся из данных исходного макета.

Относящиеся к представлениям в виде карт:

- `scale <длина изображения> <реальная длина>` ▷ устанавливает масштаб выходной карты или атласа (М, А; по умолчанию: 1 200).
- `base-scale <длина изображения> <реальная длина>` ▷ если установлено, Therion будет масштабировать карту с коэффициентом (`scale/base-scale`). Это имеет такой же эффект, как если бы карта, распечатанная в `base-scale`, была бы масштабирована к `scale`. (М, А).
- `rotate <значение>` ▷ вращает карту (М, А; по умолчанию: 0).
- `units <metric/imperial>` ▷ установить единицы измерения (М, А; по умолчанию: `metric`).
- `symbol-set <набор символов>` ▷ использование набора символов для всех символов карты, если они доступны. Помните, что имя набора символов чувствительно к регистру. (М, А).

Therion использует следующие предопределенные наборы символов:

UIS (Международный союз спелеологии)

ASF (Австралийская спелеологическая федерация)

AUT (Австрийская спелеологическая ассоциация)

5.4 CCNP (Национальный парк Карлсбадских пещер)

5.4 NZSS (набор символов Новой Зеландии)

SKBB (спелеоклуб Banská Bystrica).

- `symbol-assign <point/line/area/group/special> <тип символа> <набор символов>` ▷ отображение определенного символа в заданном наборе символов. Эта опция переопределяет опцию `symbol-set`.

Если символ имеет подтип, аргумент `<тип символа>` может иметь одну из следующих форм: `тип:подтип` или просто `тип`, который присваивает новый набор символов всем подтипам данного символа.

Следующие символы не могут использоваться с этой опцией: точечный `section` (который вообще не отображается) и все точечные и линейные надписи (`label`, `remark`, `altitude`, `height`, `passage-height`, `station-name`, `date`). Смотрите главу *Изменение макета/Настройка текстовых меток*, чтобы узнать, как изменить внешний вид надписей. (М, А).

5.3 Группа (`group`) может быть одной из следующих: `all` (все), `centerline` (нитка хода), `sections` (сечения), `water` (вода), `speleothems` (спелеотемы), `passage-fills` (рельеф хода), `ice` (лед), `sediments` (отложения), `equipment` (оборудование).

Есть два специальных символа: `north-arrow` (стрелка на север), `scale-bar` (шкала масштаба).

- `symbol-hide` `<point/line/area/group/special>` `<тип символа>` ▷ не отображать конкретный символ или группу символов.

5.4

Вы можете использовать `group cave-centerline`, `group surface-centerline`, `point cave-station`, `point surface-station` и `group text` в командах `symbol-hide` и `symbol-show`.

Используйте `flag:<entrance/continuation/sink/spring/doline/dig>` как `<тип символа>` чтобы скрыть станции с определенными флагами (например, `symbol-hide point flag:entrance`).

Может комбинироваться с `symbol-show`. (М, А).

- `symbol-show` `<point/line/area/group/special>` `<тип символа>` ▷ отображать определенный символ или группу символов. Может комбинироваться с `symbol-hide`. (М, А).

- `symbol-colo[u]r` `<point/line/area/group/special>` `<тип символа>` `<цвет>` ▷ изменить цвет конкретного символа или группы символов<sup>55</sup>. (М, А). 5.3

- `min-symbol-scale` `<масштаб>` ▷ определить минимальный `<масштаб>`, при котором точечные и линейные символы отображаются на карте. Например, при опции `min-symbol-scale M` никакие точечные или линейные символы при масштабе `S` и `XS` не будут показаны на карте. `<Масштаб>` имеет тот же формат, что и `масштаб` для точечных и линейных символов. 5.4.1

- `fonts-setup` `<tinysize>` `<smallsize>` `<normalsize>` `<largesize>` `<hugesize>` ▷ указать размер текста в точечных символах. `<normalsize>` применяется к точечному символу 'текстовая метка', `<smallsize>` к 'заметке' и всем другим точечным текстовым символам. Каждый из этих размеров может применяться к линейному текстовому символу в соответствии с ее опцией `-size`. 5.4.1

По умолчанию: 8 10 12 16 24 для шкал до 1:100; 7 8 10 14 20 для шкал до 1:200; 6 7 8 10 14 для шкал до 1:500 и 5 6 7 8 10 для шкал менее 1:500.

*Относящиеся к макету страницы:*

- `size` `<ширина>` `<высота>` `<единицы>` ▷ задает размер карты в режиме атласа. Если не указано, то размер будет рассчитан по `page-setup` и `overlap`. Применяется в режиме карты, если `page-grid` установлен в `on` (М, А; по умолчанию: 18 22,2 см).
- `overlap` `<значение>` `<единицы>` ▷ установить размер перекрытия в единицах в режиме атласа или поля карты в режиме карты (М, А; по умолчанию: 1 см).
- `page-setup` `<размеры>` `<единицы>` ▷ установить размеры страницы в таком порядке: ширина бумаги, высота бумаги, ширина страницы, высота страницы, левое поле и верхнее поле. Если не указано, то размеры будут вычислены из `size` и `overlap` (А; по умолчанию: 21 29,7 20 28,7 0,5 0,5 см).
- `page-numbers` `<on/off>` ▷ включить/выключить нумерацию страниц (А; по умолчанию: `on`).
- `exclude-pages` `<on/off>` `<список>` ▷ исключить указанные страницы из атласа. Список может содержать номера страниц, разделенные запятой или тире (для интервалов), например, `2,4-7,9,23` означает, что страницы 2, 4, 5, 6, 7, 9 и 23 должны не выводиться. Только страницы карты должны подсчитываться. (Установите `own-pages 0` и `title-pages off` чтобы получить

<sup>55</sup> Примечание: изменение цвета в настоящее время применяется к заполнению шаблона только в том случае, если (1) формат вывода PDF и (2) METAPOST версия не ниже 1.000.

корректные номера страниц после исключения). Изменения опций `own-pages` или `title-pages` не влияют на исключение страниц. (А).

- `title-pages <on/off>` ▷ включение/выключение титульных страниц перед каждой главой атласа (А; по умолчанию: `off`).
- `nav-factor <фактор>` ▷ установить коэффициент масштабирования навигатора атласа (А; по умолчанию: `30`).
- `nav-size <х-размер> <у-размер>` ▷ установить количество страниц атласа в обоих направлениях навигатора (А; по умолчанию: `2 2`).
- `transparency <on/off>` ▷ установить прозрачность для ходов (отображаются только видимые ходы) (М, А; по умолчанию: `on`).
- `opacity <значение>` ▷ установить значение непрозрачности (используется, если `transparent` установлен в `on`). Диапазон значений 0-100. (М, А; по умолчанию: `70`).
- `surface-opacity <значение>` ▷ установить прозрачность растровой поверхности (используется, если `transparent` установлен в `on`). Диапазон значений 0-100. (М, А; по умолчанию: `70`).
- `surface <top/bottom/off>` ▷ установить положение растрового изображения выше/ниже карты. (М, А; по умолчанию: `off`).
- `sketches <on/off>` ▷ отображать/скрыть растровые изображения с эскизом подвергшимся морфингу. (М, А; по умолчанию: `off`).
- `layers <on/off>` ▷ включение/отключение слоев PDF 1.5 (М, А; по умолчанию: `on`).
- `grid <off/bottom/top>` ▷ отображать/скрыть сетку (опционально могут также отображаться значения координат) (М, А; по умолчанию: `off`).
- `cs <система координат>` ▷ система координат для `origin` и `grid-origin`.
- `north <true/grid>` ▷ указать ориентацию карты по умолчанию. По умолчанию используется истинный (астрономический) север. Он игнорируется при использовании с локальной системой координат.
- `grid-origin <x> <y> <z> <единицы>` ▷ заданные координаты начала сетки (М, А).
- `grid-size <x> <y> <z> <единицы>` ▷ задать размер сетки в реальных единицах (М, А, по умолчанию равен размеру шкалы).
- `grid-coords <off/border/all>` ▷ указать, где маркировать сетку координатами. (М, А; по умолчанию: `off`).
- `origin <x> <y> <z> <единицы>` ▷ установить начало страниц атласа (М, А).
- `origin-label <х-метка> <у-метка>` ▷ установить метку для страницы атласа, которая имеет указанные координаты нижнего левого угла. Может быть либо числом, либо символом. (М, А; по умолчанию: `0 0`).
- `own-pages <число>` ▷ установить количество собственных страниц, добавленных до первой страницы автоматически созданных страниц в режиме атласа (в настоящее время требуется для правильной нумерации страниц) (А; по умолчанию: `0`).
- `page-grid <on/off>` ▷ показывать план страничных ключей (М; по умолчанию: `off`).

Относящиеся к легенде карты:

- `map-header <x> <y> <off/n/s/e/w/ne/nw/se/sw/center>` ▷ напечатать заголовок карты в месте, указанном координатами `<x> <y>`. Предопределенный заголовок карты содержит некоторую базовую информацию о пещере: имя, масштаб, направление на север, список съемщиков и т.д. Он полностью настраивается (подробнее смотрите главу *Изменение макета*). `<x>` – на восток (слева направо на странице). `<y>` – на север (вверх/вниз страницы). Диапазоны для `<x>` и `<y>` – 0-100. Нижний левый угол карты равен 0 0, верхний правый угол 100 100. Заголовок выравнивается с указанным углом или сбоку от этой точки привязки. (М; по умолчанию: 0 100 nw).
- `map-header-bg <on/off>` ▷ когда включено, фон заголовка карты заполняется цветом фона (например, для скрытия сетки). (М; по умолчанию: off).
- `map-image <x> <y> <n/s/e/w/ne/nw/se/sw/center> <имя файла>` ▷ включить изображение, указанное в `<имя файла>` на карту в месте, указанном `<x> <y>`. Для координат и деталей выравнивания смотрите спецификацию на `map-header`.
- `legend-width <n> <единицы>` ▷ ширина легенды (М, А; по умолчанию: 14 см).
- `legend <on/off/all>` ▷ отобразить список используемых символов карты в заголовке карты. Если установлено значение `all`, то отображаются все символы из текущего набора символов. (М, А; по умолчанию: off).
- `colo[u]r-legend <on/off>` ▷ показать/скрыть легенду цветов `map-fg`, когда `map-fg` установлен по высоте, скрапу или карте (М, А).
- `legend-columns <число>` ▷ указывает количество столбцов легенды (М, А; по умолчанию: 2).
- `map-comment <текст>` ▷ необязательный комментарий, отображаемый в заголовке карты (М).
- `statistics <explo/topo/carto/copyright all/off/number>` или
- `statistics <explo/topo-length on/hide/off>` ▷ отобразить некоторые основные статистические данные; если установлено значение `off`, то члены команды сортируются в алфавитном порядке; в противном случае в зависимости от их вклада в работу (М, А; по умолчанию: off).
- `scale-bar <длина> <единицы>` ▷ установить длину шкалы (М, А).
- `language <xx[_YY]>` ▷ установить язык вывода. Доступные языки перечислены на странице авторских прав. Смотрите *Приложение*, если вы хотите добавить или настроить переводы. (М, А).
- `colo[u]r <элемент> <цвет>` ▷ настроить цвет для специальных элементов карты (`map-fg`, `map-bg`, `preview-above`, `preview-below`, `label`). Цветовой диапазон 0-100 для оттенков серого, [0-100 0-100 0-100] для цветов в пространстве RGB.

Для `map-fg` вы можете использовать `altitude`, `scrap` или `map` как цвета. В этом случае карта окрашена в соответствии с высотой, скрапом или картой.

Для `map-bg` вы можете использовать `transparent` чтобы полностью исключить фон страницы.

Для подписей вы можете включать/выключать цвет `on/off`. Если `on`, то подписи окрашиваются с использованием цвета связанного скрапа.

5.4

- `debug <on/all/first/second/scrap-names/station-names/off>` ▷ рисовать скрап на разных этапах преобразования в разных цветах, чтобы увидеть, как Therion искажает данные карты. Смотрите описание `scrap` для подробностей. Точечные символы, расположение которых больше всего изменились во время преобразования, отображаются оранжевым цветом. Если `scrap-names` указано, имена скрапов отображаются для каждого скрапа, `station-names` отображает имя каждой станции съемки.
- `survey-level <N/all>` ▷ `N` – количество уровней съемки, отображаемых рядом с именем станции (M, A; по умолчанию: 0).

*Относящиеся к PDF:*

- `doc-author <текст>` ▷ установить автора документа (M, A).
- `doc-keywords <текст>` ▷ установить ключевые слова документа (M, A).
- `doc-subject <текст>` ▷ задать тему документа (M, A).
- `doc-title <текст>` ▷ установить название документа (M, A).

*Настройка:*

- `code <metapost/tex-map/tex-atlas>` ▷ Добавить/переопределить макросы `TeX` и `METAPOST`. Это позволяет пользователю настраивать различные элементы (например, пользовательские символы, макет карты и атласа в одном месте и т.д.). Для подробностей смотрите главу *Изменение макета*.
- `endcode` ▷ должен заканчивать разделы `TeX` и `METAPOST`.

## 'setup3d'

*Синтаксис:*

- `setup3d <значение>`

5.3 Установить расстояние дискретизации в метрах при генерации стен 3d-модели из кривых Безье.

## 'sketch-colors'

*Синтаксис:*

- `sketch-colors <число цветов>`

5.4 Этот параметр можно использовать для уменьшения размера растровых изображений эскиза на картах.

## 'export'

*Описание:*

Экспортирует выбранные съемки или карты.

*Синтаксис:*

- `export <тип> [ОПЦИИ]`

*Аргументы:*

- `<тип>` ▷ поддерживаются следующие типы экспорта:  
  - `model` ▷ 3D-модель пещеры;
  - `map` ▷ одностраничная 2D-карта;



atlas ▷ 2D-атлас на нескольких страницах;

cave-list ▷ сводная таблица пещер;

survey-list ▷ сводная таблица съемок;

continuation-list ▷ список возможных продолжений (перспектив);

database ▷ база данных SQL с ниткой хода.

Опции:

общие:

- `encoding/enc <кодировка>` ▷ установить кодировку вывода.
- `output/o <файл>` ▷ установить выходной файл. Если имя файла не задано, префикс "cave." используется с расширением, соответствующим выходному формату.



Если задано имя выходного файла и не указан выходной формат, формат определяется из расширения имени файла.

*model (модель):*

- **format/fmt <формат>** ▷ установить формат вывода модели. В настоящее время поддерживаются следующие форматы вывода: **loch** (собственный формат, по умолчанию), **compass** (plt файл), **survex** (3d файл), **dxg**, **esri** (3d shapefiles), **vrml**, **3dmf** и **kml** (Google Earth).
- **enable <walls/[cave/surface-]centerline/splay-shots/surface/all>** и
- **disable <walls/[cave/surface-]centerline/splay-shots/surface/all>** ▷ выбирает, какие функции экспортировать, если формат поддерживает его. Поверхность в настоящее время экспортируется только в формате **therion**.
- **wall-source <maps/centerline/all>** ▷ установить исходные данные для моделирования стен хода.

*map/atlas (карты/атлас):*

- 5.3
- **format/fmt <формат>** ▷ установить формат карты. В данный момент поддерживаются форматы **pdf**, **svg**, **xhtml**<sup>56</sup>, **survex**, **dxg**, **esri**<sup>57</sup>, **kml** (Google Earth), **xvi**<sup>58</sup> и **bbox**<sup>59</sup> для карты; **pdf** для атласа.
  - **projection <идентификатор>** ▷ уникальный идентификатор, который задает тип проекции карты (для деталей смотрите команду **scrap**).

Если карта не определена, экспортируются все скрапы в данной проекции.

Если нет скрапов с указанной проекцией, Therion отобразит нитку хода из выбранных съемок.

- **layout <идентификатор>** ▷ использовать предопределенный макет карты или атласа.
- **layout-xxx** ▷ где **xxx** означает другие опции макета. Используя это, вы можете изменить некоторые свойства макета непосредственно в команде **export**.
- **encoding/enc <кодировка>** ▷ установить кодировку вывода.

*общее для list (списков):*

- 5.4
- **format/fmt <формат>** ▷ установить формат списка возможных продолжений (перспектив). В настоящее время поддерживаются следующие форматы вывода: **html** (по умолчанию), **txt**, **kml**<sup>60</sup> и **dbf**.

*continuation-list (список перспектив):*

- 5.3
- **attributes <(on)/off>** ▷ указывает, следует ли экспортировать определенные пользователем атрибуты в таблице списка продолжений (перспектив).
  - **filter <(on)/off>** ▷ указывает, следует ли продолжения (перспективы) без комментариев/текста фильтровать.

<sup>56</sup> SVG, встроенный в XHTML-файл, который также содержит легенду.

<sup>57</sup> ESRI shapefiles. Несколько файлов записываются в каталог с указанным именем файла.

<sup>58</sup> Векторное изображение Xtherion'a. Изображения XVI могут использоваться для рисования карт в масштабе. **Scale** (принято разрешение изображения в 100 DPI) и **grid-size** макета используются в экспорте.

<sup>59</sup> Текстовый файл, содержащий географические координаты нижнего левого и верхнего правого углов области карты.

<sup>60</sup> Для списка пещер и списка продолжений (перспектив).



*cave-list* (список пещер):

- **location** <on/(off)> ▷ указать, следует ли экспортировать координаты входов пещеры в таблицу.

5.3

- **surveys** (on)/off ▷ экспортирует необработанный список пещер при установке **off**. Иначе отображается структура съемки со статистикой.

*database* (база данных):

- **format/fmt** <формат> ▷ в настоящее время поддерживаются форматы **sql** и **csv**.
- **encoding/enc** <кодировка> ▷ установить кодировку вывода.

Форматы файлов:

<i>тип экспорта</i>	<i>поддерживаемые форматы</i>
model (модель)	loch, dxf, esri, compass, survex, vrml, 3dmf, kml
map (карта)	pdf, svg, xhtml, dxf, esri, survex, xvi, kml, bbox
atlas (атлас)	pdf
database (база данных)	sql, csv
lists (списки)	html, txt, kml, dbf

5.3

5.4

## Запуск Therion'a

Теперь, после освоения данных и файлов конфигурации, мы готовы запустить Therion. Обычно это делается из командной строки в каталоге данных путем ввода

**therion**

Полный синтаксис

```
therion [-q]
        [-L]
        [-l <файл журнала>]
        [-s <исходный файл>] [-p <путь поиска>]
        [-b/-bezier]
        [-d]
        [-x]
        [-use-extern-libs]
        [<конфигурационный файл>]
```

или

```
therion [-h/-help]
        [-v/-version]
        [-print-encodings]
        [-print-environment]
        [-print-init-file]
        [-print-library-src]
        [-print-symbols]
        [-print-tex-encodings]
        [-print-xtherion-src]
```

### Аргументы:

**<конфигурационный файл>** ▷ Therion принимает только один необязательный аргумент: имя конфигурационного файла. Если имя не указано, то ищется файл **thconfig** в текущем каталоге. Если не существует файла **thconfig** (например, текущий каталог не является каталогом данных), тогда Therion выдаст сообщение об ошибке.

### Опции:

- **-d** ▷ включить режим отладки. Текущая реализация создает временный каталог с именем **thTMPDIR** (в вашем временном системном каталоге) и не удаляет временные файлы.
- **-h, -help** ▷ показать краткую справку.
- **-L** ▷ не создавайте файл журнала. Обычно Therion записывает все сообщения в файл **therion.log** в текущем каталоге.
- **-l <файл журнала>** ▷ изменить имя файла журнала.
- **-p <путь поиска>** ▷ эта опция используется для установки пути поиска (или списка путей разделенных двоеточиями), который использует Therion для поиска его исходных файлов (если он не находит их в рабочем каталоге).
- **-q** ▷ запустить Therion в тихом режиме. Он будет печатать только предупреждения и сообщения об ошибках в **STDERR**.
- **-print-encodings** ▷ выводит список всех поддерживаемых входных кодировок.
- **-print-tex-encodings** ▷ выводит список всех поддерживаемых кодировок для вывода в PDF.
- **-print-init-file** ▷ выводит файл инициализации по умолчанию. Более подробную информацию смотрите в разделе *Инициализация в Приложении*.
- **-print-environment** ▷ выводит настройки среды для Therion'a.
- **-print-symbols** ▷ выводит список всех символов карты, поддерживаемых Therion'ом в файл **symbols.xhtml**.
- **-s <исходный файл>** ▷ установить имя исходного файла.
- **-use-extern-libs** ▷ не копирует макросы **T<sub>E</sub>X** и **METAPOST** в рабочий каталог. **T<sub>E</sub>X** и **METAPOST** должны искать их самостоятельно. Используйте с осторожностью.
- **-v, -version** ▷ отображать информацию о версии.
- **-x** ▷ создает файл **'xtherion.dat'** с дополнительной информацией о XTherion'e.

## XTherion – компилятор

XTherion упрощает запуск Therion'a, особенно в системах без вывода командной строки. Окно компилятора – это окно по умолчанию в XTherion'e. Для запуска Therion'a достаточно открыть конфигурационный файл и нажать кнопку 'F9' или 'Компилировать'.

XTherion отображает сообщения Therion'a в нижней части экрана. Каждое сообщение об ошибке подсвечивается и является гиперссылкой на исходный файл, где произошла ошибка.

После первого запуска активируются дополнительные меню *Структура съемки* и *Структура карты*. Пользователь может с комфортом выбрать съемку или карту для экспорта, дважды щелкнув некоторые элементы в дереве структуры. Простой щелчок в дереве *Структура съемки* отображает основную информацию о съемке в меню *Информация*.

## Что мы получаем?

### Информационный файл

#### Файл журнала

Помимо сообщений от Therion'a и других используемых программ, файл журнала содержит информацию о вычисленных значениях магнитного склонения и конвергенции меридиана, ошибок колец и искажений скрапов.

Абсолютная ошибка кольца равна  $\sqrt{d_x^2 + d_y^2 + d_z^2}$ , где  $d_x$  разность между одинаковыми начальными и конечными точками кольца до распределения ошибок, измеренная вдоль оси координат  $x$ ; аналогично для  $y$  и  $z$ . Процентная погрешность кольца вычисляется как *абсолютная ошибка/длину кольца*. Средняя ошибка – это простое арифметическое среднее всех ошибок кольца.

Искажение скрапа вычисляется с использованием меры искажения, определенной для всех пар точек (точечные символы, точки и контрольные точки линейных символов) в скрапе. Мера рассчитывается как  $\frac{|d_a - d_b|}{d_b}$ , где  $d_b$  – это расстояние до деформирования, а  $d_a$  – расстояние до точек после деформирования. Максимальные и средние искажения скрапа вычисляются как максимальные или средние значения таких мер, применяемых ко всем парам точек.

#### XTherion

Если опция **-x** включена, то Therion выводит некоторые основные данные о каждой съемке (длина, амплитуда, длина с севера на юг, длина с запада на восток, количество замеров и станций). Эта информация отображается в окне XTherion'a *Компилятор*, в меню *Информация*, когда выбрана какая-либо съемка в меню *Структура съемки*.

#### Экспорт SQL

Экспорт SQL позволяет легко получить очень подробную информацию о нитке хода. Это текстовый файл, начинающийся с объявления таблиц (в ниже расположенном коде '?' обозначает максимальное значение, требуемое данными в колонке)

```
create table SURVEY (ID integer, PARENT_ID integer,
    NAME varchar(?), FULL_NAME varchar(?), TITLE varchar(?));
create table CENTRELINE (ID integer, SURVEY_ID integer,
    TITLE varchar(?), TOPO_DATE date, EXPLO_DATE date,
    LENGTH real, SURFACE_LENGTH real, DUPLICATE_LENGTH real);
create table PERSON (ID integer, NAME varchar(?), SURNAME varchar(?));
create table EXPLO (PERSON_ID integer, CENTRELINE_ID integer);
create table TOPO (PERSON_ID integer, CENTRELINE_ID integer);
create table STATION (ID integer, NAME varchar(?),
    SURVEY_ID integer, X real, Y real, Z real);
create table STATION_FLAG (STATION_ID integer, FLAG char(3));
create table SHOT (ID integer, FROM_ID integer, TO_ID integer,
```

```

CENTRELINE_ID integer, LENGTH real, BEARING real, GRADIENT real,
ADJ_LENGTH real, ADJ_BEARING real, ADJ_GRADIENT real,
ERR_LENGTH real, ERR_BEARING real, ERR_GRADIENT real);
create table SHOT_FLAG (SHOT_ID integer, FLAG char(3));

```

за которым следуют команды SQL 'insert'. Этот файл может быть загружен в любую базу данных SQL (после некоторой инициализации, зависящей от базы данных, которая может включать запуск SQL-сервера и подключение к нему, создание базы данных и подключение к ней. Хорошая идея – начать транзакцию перед загрузкой этого файла, если база данных не запускает транзакцию автоматически). Важно настроить кодировку базы данных так, чтобы она соответствовала той, которая указана в команде экспорта базы данных Therion'a – [export database](#).



Названия таблиц и столбцов говорят сами за себя; для неопределенных или несуществующих значений используется **NULL**. **FLAG** в таблице **SHOT\_FLAG** принимает значения **dpl** или **srf** для дублированных замеров или замеров на поверхности; в таблице **STATION\_FLAG** – **ent, con, fix, spr, sin, dol, dig, air, ove, arc** для станций с атрибутами entrance (вход), continuation (продолжение), fixed (фиксированный), spring (источник), sink (попор), doline (карстовая воронка), dig (раскоп), air-draught (ток воздуха), overhang (нависание потолка) или arch (свод), соответственно.

Примеры простых запросов:

*Список членов исследовательской группы с информацией о вкладе каждого из них в процесс съемки:*

```

select sum(LENGTH), sum(SURFACE_LENGTH), NAME, SURNAME
  from CENTRELINE, TOPO, PERSON
 where CENTRELINE.ID = TOPO.CENTRELINE_ID
    and PERSON.ID = PERSON_ID
 group by NAME, SURNAME order by 1 desc, 4 asc;

```

*Какие части пещеры были отсняты в 1998 году?*

```
select TITLE from SURVEY where ID in
  (select SURVEY_ID from CENTRELINE
   where TOPO_DATE between '1998-01-01' and '1998-12-31');
```

*Какова длина ходов промежутка между 1500 и 1550 м глубины?*

```
select sum(LENGTH) from SHOT, STATION S1, STATION S2
  where (S1.Z+S2.Z)/2 between 1500 and 1550 and
        SHOT.FROM_ID = S1.ID and SHOT.TO_ID = S2.ID.
```

## Списки – пещеры, съемки, продолжения (перспективы)

Использование [export continuation-list](#) вы получите список всех точек в нитке хода и скрапах, отмеченных<sup>61</sup> как продолжение (возможное продолжение).

[Export cave-list](#) дает вам информацию в виде таблицы об обследованных пещерах (вам нужно указать флаги [entrance](#) в ваши данные), включая длину, глубину и местоположение входа.

Подробная информация о каждой съемке дает команда [export survey-list](#). Длина включает замеры с флагами [approximate](#), но не с флагами [explored](#), [duplicate](#) или [surface](#).

## 2D-карты

### Карты для печати

Карты создаются в форматах PDF и SVG, которые могут быть просмотрены или распечатаны в большом количестве программ. Обязательно снимите флаг *Fit page to paper* (*подогнать под размер страницы*) или аналогичный вариант, если вы хотите напечатать в реальном масштабе.

В режиме атласа на каждую страницу помещается дополнительная информация: номер страницы, название карты и метка страницы.

Особенно полезны номера соседних страниц в направлениях N, S, E и W, а также в верхнем и нижнем уровнях. Отображаются также гиперссылки на границе карты, если пещера продолжается на следующей странице и в соответствующих ячейках навигатора.

Файлы PDF высоко оптимизированы – скрапы сохраняются только один раз в документе в формах XObject и отображаются на соответствующих страницах. Therion использует самые продвинутые функции PDF, такие как прозрачность и слои.

Созданные PDF-файлы могут быть необязательно пост-обработаны в таких приложениях, как pdfTeX или Adobe Acrobat – можно извлечь или изменить некоторые страницы, добавить комментарии или шифрование и т.д.

Если карта была создана с использованием данных с привязкой, то она также содержит информацию о геопривязке. Эти данные можно получить с помощью XTherion'a для создания растровых изображений с привязками (подробнее смотрите *XTherion/Дополнительные инструменты*). 5.3

<sup>61</sup> Используя атрибут [station](#) для точки нитки хода и [point continuation](#) в скрапах.

## **Карты для GIS**

Карты, созданные в форматах DXF, ESRI или KML, могут быть дополнительно обработаны в соответствующем программном обеспечении. Эти карты не содержат визуализированных символов карты.

## **Специальные карты**

Карта в формате XVI содержит нитку хода с LRUD (и, возможно, эскизом подвергнувшимся морфингу) и может быть импортирована в XTherion, чтобы служить фоном для оцифровки.

Карта в формате Survex предназначена для быстрого просмотра в Aven.

## **3D-модели**

Therion может экспортировать 3D-модель в различные форматы, помимо собственного формата. Они могут быть загружены в соответствующие программы просмотра, редактирования или трассировки для печати или дальнейшей обработки. Если формат не поддерживает отображение формы хода, то отображается только нитка хода.

## **Loch**

Loch – просмотрщик 3D-моделей, включенный в дистрибутив Therion'a. Он поддерживает, например, рендеринг с высоким разрешением и стерео-просмотр с помощью 3D-очков.



## Изменение макета PDF-карт

Эта глава чрезвычайно полезна, если вас не устраивает предопределенный макет символов карты и ее предоставление, и вы хотите адаптировать их к вашим желаниям. Тем не менее, вам нужно знать, как писать простые макросы  $\text{\TeX}$  и  $\text{\MetaPost}$ , чтобы добиться этого.

### Макет страницы в режиме атласа

Команда `layout` позволяет произвести базовую настройку страницы в режиме атласа. Это делается с помощью таких опций, как `page-setup` или `overlap`. Но нет никаких опций, которые указывали бы положение карты, навигатора и других элементов внутри области, определяемой `page-width` и `page-height`; например, почему навигатор находится ниже карты, а не справа или слева?

Существует много возможных вариантов для страницы. Вместо того, чтобы использовать еще больше опций для команды `layout`, Therion использует язык  $\text{\TeX}$  для описания других макетов страниц.

Преимущество такого подхода состоит в том, что пользователь имеет прямой доступ к расширенному механизму набора текста, не делая язык Therion сверхсложным.

Therion использует  $\text{\pdfTeX}$  с форматом *plain* для типографского набора. Поэтому вы должны быть знакомы с *plain*  $\text{\TeX}$ , если хотите определить новые макеты.

Подробнее о *plain*  $\text{\TeX}$  смотрите в

Knuth, D. E.: *The  $\text{\TeX}$ book*, Reading, Massachusetts, Addison-Wesley <sup>1</sup>1984

Для освоения  $\text{\PdfTeX}$  имеется краткое руководство

Thanh, H. T.–Rahtz, S.–Hagen, H.: *The  $\text{\pdfTeX}$  user manual*, доступное на <http://www.pdfTeX.org>

$\text{\TeX}$  макросы используются внутри `code tex-atlas` команды `layout` (подробнее смотрите главу *Обработка данных*). Основным определением Therion'a является макрос



## `\dopage`

Идея проста: для каждой страницы Therion определяет переменные `TeX` (`count`, `token` и `box registers`), которые содержат элементы страницы (карта, навигатор, название страницы и т.д.). В конце каждой страницы вызывается макрос `\dopage`. Это определяет положение каждого элемента на странице. Переопределив этот макрос, вы получите желаемый макет страницы. Без этого переопределения вы получите стандартный макет.

Ниже приведен список переменных, определенных для каждой страницы:

### *Боксы:*

- `\mapbox` ▷ бокс содержащий карту. Его ширина (высота) устанавливается в соответствии с опциями `size` и `overlap` команды `layout`.

`size_width + 2*overlap` или

`size_height + 2*overlap`, соответственно.

- `\navbox` ▷ бокс содержащий навигатор, с размерами

`size_width * (2*nav_size_x+1) / nav_factor` или

`size_height * (2*nav_size_y+1) / nav_factor`, соответственно.

Оба бокса, `\mapbox` и `\navbox`, также содержат гиперссылки.

### *Регистровые счетчики:*

- `\pointerE`, `\pointerW`, `\pointerN`, `\pointerS` содержат номер страницы соседних страниц в направлениях E, W, N и S. Если такой страницы нет, номер ее страницы равен 0.

- `\pagenum` содержит номер текущей страницы.

### *Регистры токенов:*

- `\pointerU`, `\pointerD` содержат информацию о страницах выше и ниже текущей страницы. Они состоят из одной или нескольких конкатенированных записей. Каждая запись имеет специальный формат `page-name|page-number|destination||`

Если таких страниц нет, то значение устанавливается в `notdef`.

Смотрите описание макроса `\processpointeritem` ниже, чтобы знать, как извлечь и использовать эту информацию.

- `\pagename` ▷ имя текущей карты в соответствии с опциями команды `map`.
- `\pagelabel` ▷ метка страницы, указанная опциями `origin` и `origin-label` в команде `layout`.

Следующие переменные задаются в начале документа:

- `\hsize`, `\vsize` ▷ `TeX` размеры страницы, заданные в соответствии с параметрами `page-width` и `page-height` опции `page-setup` команды `layout`. Они устанавливают нашу площадку при определении макета страницы используя макрос `\dopage`.
- `\ifpagenumbering` ▷ это значение установлено в `true` или `false` в соответствии с опцией `page-numbers` команды `layout`.

Существуют также некоторые predefined макросы, которые помогают с обработкой `\pointer*` переменных:

- `\showpointer` с одним из `\pointerE`, `\pointerW`, `\pointerN` или `\pointerS` в качестве аргумента отображает значение аргумента. Если значение равно 0, тогда ничего не отображается. Это полезно, потому что нулевое значение (нет соседней страницы) не должно отображаться.

- `\showpointerlist` с одним из аргументов `\pointerU` или `\pointerD` представляет содержание этого аргумента. Содержимое `\pointerU` и `\pointerD` смотрите выше. Для каждой записи `\showpointerlist` вызывает макрос `\processpointeritem`, который отвечает за форматирование данных.

Макрос `\showpointerlist` следует использовать без переопределения в том месте, где вы хотите отобразить содержимое своего аргумента; для форматирования пользовательских данных переопределите макрос `\processpointeritem`.

- `\processpointeritem` имеет три аргумента – `page-name` (название страницы), `page-number` (номер страницы), `destination` (место отображения), и выводит эти данные. Аргументы разделяются следующим образом:

```
\def\processpointeritem#1|#2|#3\endarg{...}
```

Пример может быть следующим:

```
\def\processpointeritem#1|#2|#3\endarg{%
  \hbox{\pdfstartlink attr {/Border [0 0 0]}%
    goto name {#3} #2 (#1)\pdfendlink}%
}
```

(обратите внимание, как использовать аргумент *destination*), или намного проще (если нам не нужны функции гиперссылки):

```
\def\processpointeritem#1|#2|#3\endarg{%
  \hbox{#2 (#1)}%
}
```

Для управления шрифтом есть следующие макросы:

- `\size[#1]` для изменения размера,
- `\color[#1 #2 #3]` для изменения цвета (значения RGB в диапазоне 0-100) и
- `\rm`, `\it`, `\bf`, `\ss`, `\si` для изменения типа отображения.

Ниже приведен список predefined настроек текстов, которые могут быть использованы в атласе.

Также есть макрос `\framed`, который принимает бокс в качестве аргумента и отображает бокс в рамке. Стиль рамки можно настроить, переопределив макрос `\linestyle`, который по умолчанию равен `1 J 1 j 1.5 w`.

Теперь мы готовы определить макрос `\dopage`. Вы можете выбрать, какой из predefined элементов использовать. Очень простой пример:

```
layout my_layout
  scale 1 200
  page-setup 29.7 21 27.7 19 1 1 cm
  size 26.7 18 cm
  overlap 0.5 cm
  code tex-atlas
    \def\dopage{\box\mapbox}
    \insertmaps
endlayout
```

который определяет размер формата А4 без навигатора и любых текстов. На странице есть только карта.

Обратите внимание на макрос `\insertmaps`. Страницы карты вставляются в его положение. Это не делается автоматически, потому что вы можете вставить некоторые другие страницы перед первой страницей карты.

Более продвинутым является определение по умолчанию макроса `\dopage`:

```
\def\dopage{
  \vbox{\centerline{\framed{\mapbox}}
    \bigskip
    \line{
      \vbox to \ht\navbox{
        \hbox{\size[20]\the\pagelabel
          \ifpagenumbering\space(\the\pagenum)\pagename
          \space\size[16]\the\qqquad}
        \ifpagenumbering
          \qqquad
          \hbox{\qqquad\qqquad
            \hss{
              \hbox to 0pt{\hss\showpointer\pointerW\hss}
              \hbox to 0pt{\hskip\showpointer\hskip\raise0.7em}
              \updownarrow1pt\hbox to 0pt{\hss$\rlap$\hss}
              \leftrightharpoon1pt\hbox to 0pt{\hss$\pointerE$\hss}
              \pointerS{\raise0.7em\showpointer\arr}}
              \hbox to 0pt{\hss\showpointer\arr\showpointer}
            }
          \vtop\vtop
          \hss{
            \def\uparrow{\$\downarrow$}
            \pointerNlist\pointerD
            \def\pointerU{\$\fi$}
            \llaplist\vss
          }
        }
      \medskip
      \scalebar
      \box
    }\showpointer
    <u74>\navbox
  }
}
```

Используя другие plain макросы T<sub>E</sub>X или примитивы T<sub>E</sub>X, можно добавить другие функции, например другой макет для нечетных и четных страниц, заголовки и колонтитулы или добавить логотип на каждую страницу.

В дополнение к страницам карты атлас содержит дополнительные области: титульная страница, основные факты о пещере, легенда с использованием символов карты и т.д.

Therion автоматически генерирует список используемых символов карты и списки лиц, которые обнаружили, исследовали и нарисовали выбранную часть пещеры. Следующие символьные регистры могут использоваться (в соответствии с необходимостью до или после макроса `\insertmaps`):

- `\explotitle`, `\topotitle`, `\cartotitle` ▷ перевод названий;
- `\exploteam`, `\topoteam`, `\cartoteam` ▷ участники (в соответствии с опциями `team`, `explo-team` команды `centreline` и опции `author` для скрапов);

- `\explodate`, `\topodate`, `\cartodate` ▷ соответствующие даты;
- `\comment` ▷ задано в соответствии с опцией `map-comment` команды `layout`;
- `\copyrights` ▷ устанавливается в соответствии с вариантами авторского права для съемок и других объектов;
- `\cavename` ▷ имя экспортируемой карты, устанавливается в соответствии с опцией `-title` экспортируемой карты;
- `\cavelength`, `\cavedepth` ▷ приблизительная длина и глубина отображаемой карты;
- `\cavelengthtitle`, `\cavedepthtitle` ▷ переведенные метки;
- `\cavemaxz`, `\caveminz` ▷ значение высоты max/min; 5.4
- `\thversion` ▷ текущая версия Therion'a; 5.4
- `\currentdate` ▷ текущая дата; 5.4
- `\outcscode`, `\outsname` ▷ вывести код и наименование системы координат; 5.4
- `\northdir` ▷ 'true' или 'grid'; 5.4
- `\magdecl` ▷ магнитное склонение в градусах; 5.4
- `\gridconv` ▷ конвергенция сетки меридиана в градусах. 5.4

Макрос `\atlastitlepages` объединяет большинство перечисленных токенов, чтобы получить простой, предварительно отформатированный, вывод страницы в виде атласа.

Для отображения легенды есть макросы:

- `\iflegend` ▷ условный; true если опция `legend` команды `layout` была установлена в `on` или `all`;
- `\legendtitle` ▷ регистровый токен, содержащий переведенное название легенды;
- `\insertlegend` ▷ макрос для вставки изображений символов легенды с переведенными описаниями в указанное количество столбцов (в соответствии с опцией `legend-columns` макета);
- `\formattedlegend` ▷ объединяет все три приведенные выше команды, чтобы получить предварительно отформатированную легенду с заголовком и символами в двух<sup>62</sup> столбцах, если опция `legend` установлена в `on`.

Стрелка направления на север и шкала масштаба могут отображаться с использованием:

- `\ifnortharrow` ▷ условный; true, если проекция карты – это план, а символ стрелки на север не скрыт в `layout`;
- `\ifscalebar` ▷ условный; true, если шкала масштаба не скрыта;
- `\northarrow` ▷ PDF-форма со стрелкой на север;
- `\scalebar` ▷ PDF-форма со шкалой масштаба.

Есть макрос общего назначения для набора текста в нескольких столбцах<sup>63</sup>:

- `\begmulti <i>`, `\endmulti` ▷ текст между этими макросами набирается в столбцы `<i>`.

Пример создания атласа со списками съемщиков и т.д. за которыми следуют карты и с легендой в конце:

<sup>62</sup> Значение по умолчанию; устанавливайте опцию `legend-columns` макета, чтобы указать количество столбцов.

<sup>63</sup> Не используйте с легендой карты оформленной в несколько столбцов с помощью опции `legend-columns` макета.

code tex-atlas \atlastitlepages

\insertmaps

\formattedlegend

### Макет страницы в режиме карты

В режиме карты можно использовать множество predefined переменных, которые описаны в предыдущей главе:

\cavename, \comment, \copyrights, \explotitle, \topotitle, \cartotitle,  
\exploteam, \topoteam, \cartoteam, \explodate, \topodate, \cartodate,  
\cavelength, \cavedepth, \cavelengthtitle, \cavedepthtitle, \cavemaxz,  
\caveminz, \thversion, \currentdate, \outscscode, \outcsname, \northdir,  
\magdecl, \gridconv, \ifnortharrow, \ifscalebar, \northarrow, \scalebar,  
\iflegend, \legendtitle, \insertlegend, \begmulti <i>, \endmulti,  
\formattedlegend, \legendcolumns.

Чтобы разместить их где-нибудь на странице карты, вам нужно определить макрос `\maplayout` в разделе `code tex-map` команды `layout`. Он должен содержать один или несколько `\legendbox` вызовов. Макрос `\legendbox` имеет четыре параметра: координаты в диапазоне 0–100, спецификация выравнивания (N, E, S, W, NE, SE, SW, NW или C) и отображаемый контент.

Простой пример:

```
\def\maplayout{
  \legendbox{0}{100}{NW}{\northarrow}
}
```

который отображает стрелку на север в верхнем левом углу листа карты.

Для удобства пользователя существует символьный регистр `\legendcontent`. Он содержит предварительно отформатированное имя пещеры, стрелку на север, шкалу масштаба, команды исследователей, съемщиков и авторов карты, комментарии, авторские права и легенду. (`\legendcontent` также используется в определении макета карты по умолчанию:

```
\def\maplayout{\legendbox{0}{100}{NW}{\the\legendcontent}}).
```

Ширина вышеуказанного текста может быть скорректирована с помощью регистра `\legendwidth` (его значение по умолчанию задано опцией `legend-width` эскиза). Цвет и размер текстов в преформатированной легенде можно легко изменить, используя символьные регистры `\legendtextcolor`, `\legendtextsize`, `\legendtextsectionsize` и `\legendtextheadersize`, например для большого синего текста:

```
code tex-map
  \legendwidth=20cm
  \legendtextcolor={\color{0 0 100}}
  \legendtextsize={\size[20]}
  \legendtextheadersize={\size[60]}
```

Можно отобразить всю карту, обрамленную установкой регистра `\framethickness` в положительное значение, например `0.5mm`.

### Настройка текстовых меток

Начиная с версии 5.4.1 вы можете использовать опцию `fonts-setup` макета вместо макроса METAPOST `fonts_setup()`.

## Новые символы карты

Команда `layout` Therion'a позволяет легко переключаться между различными предопределенными наборами символов карты. Если такой символ или набор символов вам не нужен, можно создать новые символы карты.

Однако для этого требуется знание языка METAPOST, который используется для визуализации карты. Это описано в:

Hobby, J. D.: *A User's Manual for MetaPost*, доступно здесь:  
<http://cm.bell-labs.com/cm/cs/ctr/162.ps.gz>

Пользователь также может воспользоваться исчерпывающей ссылкой на язык METAFONT, который очень похож на METAPOST:

Knuth, D. E.: *The METAFONTbook*, Reading, Massachusetts, Addison-Wesley  
1986

Новые символы могут быть определены в разделе `code metapost` команды `layout`. Это упрощает добавление новых символов во время выполнения. Также можно добавлять символы для постоянного использования, компилируя их в исполняемый файл Therion (смотрите Приложение для инструкций, как это сделать).

Каждый символ должен иметь уникальное имя, состоящее из следующих элементов:

- одна из букв 'p', 'l', 'a', 's' для точки, линии, области или специальных символов, соответственно;
- символ подчеркивания;
- тип символа, указанный в главе *Формат данных* с удалением всех тире;
- если символ имеет подтип, добавьте символ подчеркивания и подтип;
- символ подчеркивания;
- идентификатор набора символов в верхнем регистре.

Пример: стандартное имя для символа 'water-flow' с подтипом 'permanent' в наборе 'MY' – `p_waterflow_permanent_MY`. Стандартное имя для пользовательских типов символов не должно включать идентификатор набора символов, например `p_u_bat`.

Каждый новый символ должен быть зарегистрирован вызовом макроса:

```
initsymbol("<стандартное имя символа>");
```

если он не скомпилирован в исполняемый файл Therion'a.

Существует четыре предопределенных ручки *PenA* (самый толстый) ... *PenD* (самый тонкий), которые должны использоваться для всех чертежей. Для рисования и заполнения используйте команды `thdraw` и `thfill` вместо команд METAPOST `draw` и `fill`.

Доступны также следующие переменные:

- булевская `ATTR__shotflag_splay`, `ATTR__shotflag_duplicate`, `ATTR__shotflag_approx` > устанавливает линию съемки;
- булевская `ATTR__stationflag_splay` > установлено в 'true' для боковых замеров крайних станций;

5.4



- булевская `ATTR__scrap_centerline` ▷ установлено в 'true' для скрапов созданных по нитке хода;
- булевская `ATTR__elevation` ▷ устанавливается в 'true' для разреза (разрез-развертки), в 'false' для проекции плана;
- числовая `ATTR__height` ▷ глубина колодца (pit) или входного колодца (wall:pit);
- строковая `ATTR__id` ▷ содержит текущий идентификатор объекта ID;
- строковая `ATTR__survey` ▷ содержит текущее название съемки;
- строковая `ATTR__scrap` ▷ содержит текущее название скрапа;
- изображение `ATTR__text` ▷ содержит набранный текст, например для точечного символа – продолжение (continuation);
- строковая `NorthDir` ▷ 'true' или 'grid';
- числовая `MagDecl` ▷ магнитное склонение в градусах;
- числовая `GridConv` ▷ конвергенция сетки магнитного меридиана в градусах.

## Точечные символы

Точечные символы определяются как макросы с использованием команд `def ... enddef`. Большинство определений точечных символов имеют четыре аргумента: position (положение, пара), rotation (вращение, числовое), scale (масштаб, числовое) и alignment (выравнивание, пара). Исключения составляют *section (сечение)*, который не имеет визуального представления; все *labels (погнуси)*, которые требуют специальной обработки, как описано в предыдущей главе; и *station (станция)*, которая принимает только один аргумент: position (положение, пара).

Все точечные символы рисуются в локальных координатах с единицей длины *u*. Рекомендуемые диапазоны:  $\langle -0.5u, 0.5u \rangle$  в обоих осях. Символ должен быть центрирован в начале координат. Для окончательной карты все чертежи трансформируются, как указано в переменной трансформации *T*, поэтому перед ее рисованием необходимо установить эту переменную.

Обычно это делается в два этапа (предположим, есть четыре аргумента: *P*, *R*, *S*, *A*):

- устанавливается переменная *U* (пара) символа в  $\left(\frac{width}{2}, \frac{height}{2}\right)$  для правильного выравнивания. Аргумент alignment (выравнивание) *A* является парой значений, представляющих собой отношения  $\left(\frac{shift_x}{U_x}\right)$  и  $\left(\frac{shift_y}{U_y}\right)$ .

Следовательно, `aligned A` означает `shifted (xpart A * xpart U, ypart A * ypart U)`.

- устанавливается переменная трансформации *T*  
`T:=identity aligned A rotated R scaled S shifted P;`

Для рисования и заполнения используйте команды `thdraw` и `thfill` вместо команд `METAPOST draw` и `fill`. Они автоматически обрабатывают трансформацию *T*.

Пример может быть следующим:



```
def p_entrance_UIS (expr P,R,S,A)=
  U:=(.2u,.5u);
  T:=identity aligned A rotated R scaled S shifted P;
  thfill (-.2u,-.5u)-(0,.5u)-(.2u,-.5u)-cycle;
enddef;
initsymbol("p_entrance_UIS");
```

## Линейные символы

Линейные символы отличаются от точечных символов тем, что нет локальной системы координат. Каждый линейный символ получает *путь* в абсолютных координатах в качестве первого аргумента. Поэтому необходимо установить переменную *T* в *identity* перед рисованием.

Следующие символы принимают дополнительные аргументы:

- *arrow* (стрелка) ▷ числовая: 0 – нет стрелок, 1 – стрелка в конце, 2 – в начале, 3 – с обоих концов;
- *contour* (контур) ▷ текст: список точек, которые получают галочку или может принимать одно из значений: –1, –2 или –3 для неопределенной галочки, галочки в середине или без галочки, соответственно;
- *section* (сечение) ▷ текст: список точек, которые получают стрелку ориентации или –1, чтобы указать стрелки;
- *slope* (склон) ▷ числовая: 0 – нет границы, 1 – есть граница; текст: список триплет (точка, направление, длина).

Пример использования:

```
def l_wall_bedrock_UIS (expr P) =
  T:=identity;
  pickup PenA;
  thdraw P;
enddef;
initsymbol("l_wall_bedrock_UIS");
```

## Символы областей

Области похожи на линии: они принимают только один аргумент – *путь* в абсолютных координатах.

Вы можете заполнить их тремя способами:

- заполнить единую или рандомизированную сетку во временном изображении (имеющем размеры *bbox path*) с некоторыми точечными символами; закрепите его в соответствии с *path* и добавьте в *currentpicture*;
- заполните *path* сплошным цветом;
- заполните *path* с предопределенным шаблоном с использованием *withpattern*.

Шаблоны определяются с использованием того же пользовательского интерфейса (без макроса *patterncolor*), как описано в статье:

Bolek, P.: "METAPOST and patterns," *TUGboat*, 3, XIX (1998), pp. 276–283, доступна тут: <https://www.tug.org/TUGboat/Articles/tb19-3/tb60bolek.pdf>

Вы можете использовать стандартный METAPOST *draw* и аналогичные макросы без установки переменной *T* в определениях шаблонов.

Пример того, как определять и использовать шаблоны:

```

beginpattern(pattern_water_UIS);
  draw origin-10up withpen pensquare scaled (0.02u);
  patternxstep(.18u);
  patterntransform(identity rotated 45);
endpattern;

def a_water_UIS (expr p) =
  T:=identity;
  thclean p; thfill p withpattern pattern_water_UIS;
enddef;
initsymbol("a_water_UIS");

```

### **Специальные символы**

В настоящее время существуют два специальных символа: шкала масштаба и стрелка на север. Оба экспериментальные и могут быть изменены.

1. Когда выдающийся, но пожилой ученый утверждает, что что-то возможно, он почти наверняка прав. Когда он утверждает, что что-то невозможно, он, вероятно, ошибается.
2. Единственный способ обнаружить пределы возможного – это отправиться в путь к невозможному.
3. Любая достаточно развитая технология неотличима от магии.

– С. Clarke, 1973

## Приложение

### Компиляция

Если вы хотите скомпилировать Therion из исходников и запустить его, вам нужны (первые три необходимы только во время компиляции):

- GNU C/C++ компилятор.
- GNU make.
- Perl.
- Python 2.7 или 3.
- Tcl/Tk 8.4.3 или новее (<https://www.tcl.tk>) с набором виджетов *BWidget* (<https://sourceforge.net/projects/tcllib/>) и необязательным расширением *tkImg* (<https://sourceforge.net/projects/tkimg/>).
- T<sub>E</sub>X дистрибутив с plain форматом T<sub>E</sub>X, pdfT<sub>E</sub>X, и METAPOST (<https://www.tug.org>).
- LCDF Typetools пакет (<https://www.lcdf.org/type/>).
- ImageMagick дистрибутив с утилитами *convert* и *identify*, если вы хотите использовать деформирование эскизы съемок.
- *ghostscript*, если вы хотите создать откалиброванные изображения карт с привязкой к географическим картам.

Для компиляции Loch вам необходимы:

- freetype 2 или новее с поддержкой freetype-config;
- wxWidgets 2.6 или новее с поддержкой wx-config;
- VTK 5.0 или новее;
- libjpeg, libpng, zlib.

Все программы (за исключением пакетов BWidget и tkImg) обычно включаются в дистрибутивы Linux, Unix или MacOS X. Для Windows необходимы MinGW и MSYS (<http://www.mingw.org>). Эти дистрибутивы содержат GNU make и GCC. Кстати, почему бы не использовать предварительно скомпилированную версию для Windows?

## Быстрый старт

- распаковать исходники дистрибутива [therion-5.\\*.tar.gz](#);
- `cd therion`;
- `make config-macosx` или `make config-win32`, если вы используете MacOS X или Windows, соответственно;
- `make`;
- `sudo make install`.

## Руководство для хакеров

### *Make параметры*

Файл Therion'a *makefile* может принимать некоторые необязательные опции:

- `config-linux`, `config-macosx`, `config-win32` ▷ настроить Therion для конкретной платформы. Linux по умолчанию;
- `config-release`, `config-oxygen`, `config-ozone` ▷ установить уровень оптимизации для компилятора C++ (`none`, `-O2` или `-O3`);
- `config-debug` ▷ полезно перед отладкой программы;
- `install` ▷ установить Therion;
- `clean` ▷ удалить все временные файлы.

### 5.4 Кросс-компиляция для Windows

Therion поддерживает компиляцию исполняемых файлов Win32 в Linux с использованием кросс-компилятора MXE (<http://mxe.cc>).

- установите следующие static/win32-пакеты (`i686-w64-mingw32.static-*`) в директорию `/usr/lib/mxe/`: `binutils`, `bzip2`, `expat`, `freetype-bootstrap`, `gcc`, `gettext`, `glib`, `harfbuzz`, `jpeg`, `libiconv`, `libpng`, `tiff`, `vtk`, `wxwidgets`, `xz`, `zlib`;
- измените PATH: `export PATH=/usr/lib/mxe/usr/bin:$PATH`;
- `cd therion`;
- `make config-win32cross`;
- `make`.

### *Добавление новых переводов*

Therion поддерживает перевод текстовых меток на картах. Предположим, вы хотите добавить новый язык `xx`:

- запустите `'perl process.pl export xx'` в подкаталоге Therion'a `'thlang'`; Это создаст файл `texts_xx.txt`. Этот файл имеет кодировку UTF-8;
- отредактируйте файл `texts_xx.txt`. Добавьте свои переводы в строки, начинающиеся с `'xx:'`;
- запустите `make update`;
- скомпилируйте Therion.

### Добавление новых кодировок

Хотя кодировка Unicode UTF-8 содержит все символы, которые Therion может использовать, возможно вам будет неудобно использовать ее. В этом случае можно добавить поддержку любой 8-битной кодировки для текстовых входных файлов. Скопируйте файл перевода в [thchencdata](#) каталог; добавьте его имя в хэш-код 'ifiles' в начале скрипта Perl [generate.pl](#); запустите его и перекомпилируйте Therion.

Файл перевода должен содержать два шестнадцатеричных символа (первый в 8-разрядной кодировке, второй в Unicode) в каждой строке. Можно добавлять комментарии за символом '#'.

### Добавление новых T<sub>E</sub>X кодировок

Легко добавить новые кодировки для вывода 2D-карты<sup>64</sup>. Скопируйте соответствующий файл сопоставления кодировки с расширением \*.enc в [texenc/encodings](#), запустите скрипт Perl [mktexenc.pl](#) расположенный в каталоге [texenc](#) и перекомпилируйте Therion.

Therion использует те же файлы кодировки, что и программа [afm2tfm](#) из дистрибутива T<sub>E</sub>X, который имеет тот же формат, что и векторная кодировка в шрифте PostScript. Вы можете найти более подробную информацию в главе *6.3.1.5 Формат файла кодировки* в документации к программе Dvips.

### Генерация новых header-файлов T<sub>E</sub>X и METAPOST

Therion использует T<sub>E</sub>X и METAPOST для 2D-визуализации карт и верстки. Предопределенные макросы компилируются в исполняемый файл Therion'a и копируются в рабочий каталог непосредственно перед запуском METAPOST и T<sub>E</sub>X (если не используется опция [-use-extern-lib](#)s). Команда [layout](#) позволяет изменять некоторые макросы в файле конфигурации во время выполнения.

Тем не менее, возможно внесение постоянных изменений в файлы макросов. После изменения файлов в каталогах [mpost](#) и [tex](#) необходимо запустить Perl-скрипты [genmpost.pl](#) и [gentex.pl](#), которые генерируют header файлы C++, и снова компилируют исполняемый файл Therion'a.

## Переменные среды

Therion считывает следующие переменные среды:

- [THERION](#) ▷ [не требуется] путь поиска для файла(ов) (x)therion.ini;
- [HOME](#) ([HOMEDRIVE](#) + [HOMEPATH](#) на WinXP) ▷ [не требуется, но обычно присутствует в вашей системе] путь поиска для файла(ов) (x)therion.ini;
- [TEMP](#), [TMP](#) ▷ системный временный каталог, где Therion хранит временные файлы (в каталоге с именем [th\\$PID\\$](#), где [\\$PID\\$](#) является идентификатором процесса), если только [tmp-path](#) указан в файле инициализации.

Проконсультируйтесь с документацией вашей ОС о том, как их установить.

---

<sup>64</sup> Этот раздел относится к выбору шрифтов старого стиля с использованием команды [tex-fonts](#) в файле инициализации и устарела при использовании команды [pdf-fonts](#). 5.3

## Файлы инициализации

Системно-зависимые параметры Therion'a и XTherion'a указаны в файле `therion.ini` или `xtherion.ini`, соответственно. Они ищутся в следующих каталогах:

- в UNIX: `., $THERION, $HOME/.therion, /etc, /usr/etc, /usr/local/etc`;
- в Windows: `., $THERION, $HOME\therion, <директория установки Therion'a>, C:\WINDOWS, C:\WINNT, C:\Program Files\Therion`

## Therion

Если файл не найден, Therion использует настройки по умолчанию. Если вы хотите их вывести, используйте опцию `-print-init-file`. Файл инициализации читается как любой другой файл. Пустые строки или строки, начинающиеся с '#', игнорируются, строки, заканчивающиеся на обратную косую черту, продолжаются в следующей строке. В настоящее время поддерживаются команды инициализации:

- `loop-closure <therion/survex>`

По умолчанию используется `survex`, если он установлен, иначе используется `therion`;

- `encoding-default <имя кодировки>`

Устанавливает стандартную выходную кодировку (в настоящее время не используется);

- `encoding-sql <имя кодировки>`

Устанавливает стандартную выходную кодировку для экспорта SQL;

- `language <xx[_YY]>`

Язык вывода по умолчанию. Список доступных языков смотрите на странице авторских прав;

- `units <metric/imperial>`

Устанавливает единицы по умолчанию;

- `mpost-path <путь файла>`

Устанавливает полный путь к исполняемому файлу METAPOST, если Therion не сможет его найти ("`mpost`" является значением по умолчанию);

- `mpost-options <текст>`

Устанавливает опции METAPOST;

- `pdftex-path <путь файла>`

Устанавливает полный путь к исполняемому файлу pdfTeX, если Therion не сможет его найти ("`pdfetex`" является значением по умолчанию);

- `identify-path <путь файла>`

Устанавливает полный путь к идентификатору ImageMagick, если Therion не может его найти ("`ident`" является значением по умолчанию);

- `convert-path <путь файла>`

Устанавливает полный путь к исполняемому файлу ImageMagick, если Therion не может его найти ("`convert`" является значением по умолчанию);

- `source-path <каталог>`

Путь к файлам данных и конфигурации. Используется в основном для system-wide grades и определений макетов;

- `tmp-path <каталог>`

Путь, где должен быть создан временный каталог;

- `tmp-remove <команда ОС>`

Системная команда для удаления файлов из временного каталога;

- `tex-env <on/off>`

Работает только в Windows. Когда установлено значение `off` (по умолчанию), Therion временно очищает все переменные среды, связанные с  $\TeX$ . Полезно, если в вашей системе установлен другой дистрибутив  $\TeX$ , который настроил переменные среды, которые могут запутать программы  $\TeX$  и METAPOST, поставляемые с дистрибутивом Therion для Windows.

Установите в `on`, если вы используете другой дистрибутив  $\TeX$  для обработки карт;

- `text <ID языка> <текст Therion'a> <свой текст>`

С помощью этой опции вы можете изменить любой текст перевода по умолчанию на выходе. Список текстов и доступных переводов смотрите в файле `thlang/texts.txt`;

- `cs-def <id> <proj4def>`

Определяет новую систему координат `<id>` используя синтаксис Proj4;

- `pdf-fonts <rm> <it> <bf> <ss> <si>`

Настройка шрифтов, которые будут использоваться в PDF-картах. За командой должны следовать пути, указывающие, где в вашей системе расположены обычные, курсивные, полужирные, без засечек и наклонные без засечек шрифты. Поддерживаются шрифты TrueType и OpenType.

Для использования этой команды Therion требует установки LCDF Typetools в вашей системе. Пример:

```
pdf-fonts "/usr/share/fonts/Serif.ttf" \
          "/usr/share/fonts/Serif-Italic.ttf" \
          "/usr/share/fonts/Serif-Bold.ttf" \
          "/usr/share/fonts/Sans.ttf" \
          "/usr/share/fonts/Sans-Oblique.ttf"
```

- `otf2pfb <on/off>`

Когда установлено значение `on` (по умолчанию), шрифты OpenType, используемые в `pdf-fonts`, преобразуются в шрифты PFB, если они основаны на PostScript. Некоторая информация теряется в формате PFB, но есть преимущество в том, что `pdfTeX` может встраивать часть шрифтов PFB (в отличие от шрифтов OpenType, которые должны быть встроены полностью);

- `tex-fonts <кодировка> <rm> <it> <bf> <ss> <si>`

Исходный и более сложный способ настройки шрифтов для PDF-карт. Вам нужно явно указать кодировку (не более 256 символов из шрифта, который будет использоваться). Список поддерживаемых в настоящее время кодировок выводится опцией `-print-tex-encodings` в командной строке. Та же кодировка должна использоваться при генерации  $\TeX$  метрик (`*.tfm` файлов) для этих шрифтов (например, с использованием

5.4

5.3

5.3



программы afm2tfm), и эта кодировка должна быть явно указана в файле карты в pdfTeX. Единственным исключением является базовый набор шрифтов Computer Modern, который использует 'сырую' кодировку. Эта кодировка не нуждается в указывании в файле карты pdfTeX.

За кодировкой должны следовать пять спецификаций шрифтов для обычных, курсивных, полужирных, без засечек и наклонных без засечек. Значение по умолчанию: `tex-fonts raw cmr10 cmti10 cmbx10 cmss10 cmssi10`.

Пример использования других шрифтов (например, TrueType Palatino в кодировке xl2, кодировка, полученная из ISO8859-2). Запустите:

```
ttf2afm -e xl2.enc -o palatino.afm palatino.ttf
afm2tfm palatino.afm -u -v vpalatino -T xl2.enc
vptovf vpalatino.vpl vpalatino.vf vpalatino.tfm
```

Вы получите файлы `vpalatino.vf`, `vpalatino.tfm` и `palatino.tfm`. Добавьте строчку

```
palatino <xl2.enc <palatino.ttf
```

в pdfTeX файл карты. То же самое должно быть сделано для курсивных и полужирных шрифтов и соответствующих шрифтов без засечек и наклонных без засечек. Если вы ленитесь, попробуйте:

```
tex-fonts xl2 palatino palatino palatino palatino
```

(Мы должны использовать виртуальный шрифт `vpalatino` вместо `palatino`, который не содержит кернинга или лигатуры, но pdfTeX не поддерживает команду `\pdfincludechars` на виртуальных шрифтах. Будет улучшен).

Если вы хотите добавить некоторые не поддерживаемые кодировки, прочитайте главу *Компиляция / Руководство хакера*;

- `tex-fonts-optional <кодировка> <rm> <it> <bf> <ss> <si>`

Подобно смотрите в `tex-fonts`, но дополнительно проверяется, установлены ли в системе шрифты TeX. Ничего не происходит, если какой-либо из указанных шрифтов отсутствует.

Этот параметр используется по умолчанию для чешских/словацких и кириллических шрифтов, чтобы избежать ошибок METAPOST для систем без этих шрифтов.

Поскольку тест занимает некоторое время (экземпляр pdfTeX запущен), вы можете полностью отключить поведение по умолчанию, установив `tex-fonts` в файле INI.

## XTherion

Файл инициализации для XTherion на самом деле является сценарием Tcl, обрабатываемыми при запуске XTherion'a. Файл комментирован; смотрите комментарии для подробностей.

## Ограничения

- размер скрапа  $\triangleright \approx 2.8 \times 2.8$  м в выходном масштабе (ограничение METAPOST);
- размер страницы  $\triangleright$   
PDF-карта или атлас:  $\approx 5 \times 5$  м (ограничение pdfTeX)

SVG-карта: без ограничений;

- количество скрапов  $\triangleright$  примерно 500–6000, в зависимости от частоты поперечных сечений

текущий предел METAPOST:  $4(\text{скрапы} + \text{сечения}) < 4096$  (может быть произвольно увеличено)

предел pdfTeX:  $2 \times \text{страницы} + \text{изображения} + \text{шаблоны} + 6(\text{скрапы} + \text{сечения}) < 32500$

### Примеры данных

Следующий простой пример иллюстрирует базовое использование команд Therion'a:

encoding utf-8

survey main -title "Тестовая пещера"

```
survey first
  centreline
    units compass grad
    data normal from to compass clino length
          1      2 100      -5    10
  endcentreline
endsurvey
```

```
survey second -declination [3 deg]
  centreline
    calibrate length 0 0.96
    data normal from to compass length clino
          1      2 0        10    +10
  endcentreline
endsurvey
```

```
centreline
  equate 2@first 1@second
endcentreline
```

# скрапы обычно находятся в отдельных \*.th2 файлах  
scrap s1 -author 2004 "Команда Therion'a"

```
point 763 746 station -name 2@second
point 702 430 station -name 2@first
point 352 469 station -name 1@first
point 675 585 air-draught -orientation 240 -scale large
```

```
line wall -close on
  287 475
  281 354 687 331 755 367
  981 486 846 879 683 739
  476 561 293 611 287 475
endline
```

endscrap

map m1 -title "Тестовая карта"

```

s1
endmap

endsurvey

Соответствующий конфигурационный файл может быть:

encoding utf-8
source test

layout l1
  scale 1 100
  layers off
endlayout

select m1@main

export model -fmt survex
export map -layout l1

```

Если вы сохраните файл данных как 'test.th' и конфигурационный файл как 'thconfig', вы можете обработать их с помощью Therion'a.

## История изменений

### • 1999

Октябрь: первые конкретные идеи.

Ноябрь: начало программирования (скрипты Perl и макросы METAPOST).

Декабрь, 27: Therion впервые компилирует простую карту в формате PostScript (32 Кбайт Perl и 7 Кбайт METAPOST и T<sub>E</sub>X исходного кода). Модель деформации карты существенно отличалась от текущей (позиции характеристик были относительно конкретного съемного замера, а не позиции всех станций в скрапе). Эта версия уже включала некоторые интересные функции, такие как *функции трансформации*, которые позволяли определять спецификацию входного формата для данных съемки или разбивать большие карты на несколько листов.

Декабрь, 30: первая веб-страница (с примерами данных, но без исходного кода).

### • 2000

Январь: xthedit (Tcl/Tk) – графический интерфейс для Therion'a.

Февраль, 18: начало перепрограммирования (Perl).

Апрель, 1: первая гиперссылка PDF-карты/атласа.

Август: эксперименты с PDF, pdfT<sub>E</sub>X и METAPOST

### • 2001

Ноябрь: начало повторной реализации с нуля: Therion (C++ с некоторыми скриптами Perl, унаследованными от предыдущей версии); понятие скрапа; интерактивный редактор 2D-карт ThEdit в качестве замены xthedit (Delphi).

Декабрь: ThEdit впервые экспортирует простую карту.

### • 2002

Март: Therion 0.1 – Therion способен обрабатывать данные съемки (центральной линии) пещеры Мертвых летучих мышей. XTherion – текстовый редактор, предназначенный для Therion'a (Tcl/Tk).

Июль, 27: Therion 0.2 – Therion впервые компилирует простую карту (состоящую из двух скрапов) (800 КБ исходного кода).

Август: XTherion расширен до редактора 2D-карт (в качестве замены ThEdit).

Сентябрь: Therion собирает первую реальную и сложную карту пещеры. XTherion использует компилятор.

- **2003**

Март: первая версия Therion Book завершена.

Апрель: Therion включен в Debian GNU/Linux.

Июнь: все сценарии Perl переписанные на C++, Therion – это одна исполняемая программа (хотя использует Surver и T<sub>E</sub>X).

- **2004**

Март: Therion 0.3 – Therion экспортирует трехмерную модель, созданную с 2D-карт. Алгоритм замыкания колец включен в Therion.

- **2006**

Октябрь: Therion 0.4 – Новый 3D-просмотрщик (Loch).

- **2007**

Февраль: Therion 0.5 – Поддержка морфинга эскизов растровых изображений.

## **Будущее**

Хотя Therion уже используется для создания карт, есть много новых функций, которые необходимо реализовать:

### **Главная**

- информация о закрытии колец в SQL.

### **2D-карты**

- завершить предопределенные наборы символов;
- генерировать регистры для атласа;
- использовать MPlib вместо METAPOST

### **3D-модели**

- улучшить моделирование ходов.

### **XTherion**

- улучшить возможности 2D-редактирования.

## **Loch**

- цветовые схемы;
- дерево съёмки для выбора подсекций для отображения;
- пространственная фильтрация (например, отсечение плоскостями);
- поддержка нескольких поверхностей.

## **Лабиринт**

- полностью новый графический интерфейс в далеком будущем (смотрите <https://labyrinth.speleo.sk>)

*Нет царского пути в геометрии.  
Μὴ εἶναι βασιλικὴν ἀτραπὸν ἐπὶ γεωμετρίας.*

—Евклид, 3 век д.н.э.

*Нет царского пути, но есть путь.*

—Фредерик П.Брукс, 1987

## Изучение примеров

Эта глава автоматически создается из каталога примеров в исходном коде Therion'a.

### Рисование карт в Therion'e



### Список пещер

При съемке пещер на какой-то большой территории вам часто нужно создать список всех пещер на этой территории. В наборе данных, "пещера" может быть определена несколькими способами, в зависимости от уровня пещерной съемки.

Если у вас есть станция на входе с известными координатами (например, используя GPS или стандартные методы съемки), все, что вам нужно сделать, это назначить флаг **entrance** (вход)

### station cave-1 "Не снятая пещера" entrance

Если за этим входом есть какой-то разведанный проход, но вы не обследовали его должным образом, вы должны добавить также флаги `continuation` (продолжение) и `explored` (исследование), за которым следует исследуемая длина. Исследуемые, но не отснятые хода также являются частью сводной статистики.

### station cave-2 "Не снятая, но исследованная" entrance \ continuation explored 80m

Если вы как следует отсняли какую-то пещеру, ее данные обычно находятся в некоторых `survey`. Съемка становится пещерой, если у нее есть станция с опцией `-entrance`. Пример:

### survey small\_cave -title "Пещера Туннель" -entrance 0

Если съемка содержит данные, но не указан главный входа, то съемка не считается пещерой, как единица организации данных и, таким образом, не присутствует в списке пещер. В итоговой таблице присутствуют только съемки, содержащие пещеры (т.е. входные станции или съемки с основным входом).

После того, как пещеры в ваших данных были правильно определены, вы можете создать таблицу, используя

`export cave-list -o caves.html`

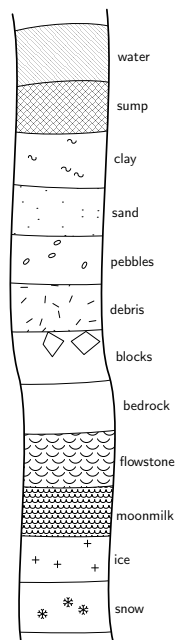
Итоговая таблица выглядит так:

Наименование	Длина	Глубина	Исследовано	Высота н.у.м.
Пещера Туннель	25	7		1244.0
Не снятая пещера				1234.0
Не снятая, но исследованная			80	1256.0



## Символы площади

Therion area types



## Морфинг эскиза

Оригинальный эскиз съемки:



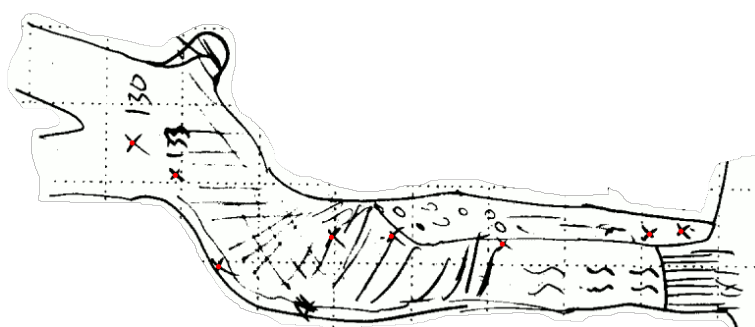
Изображение после морфинга:



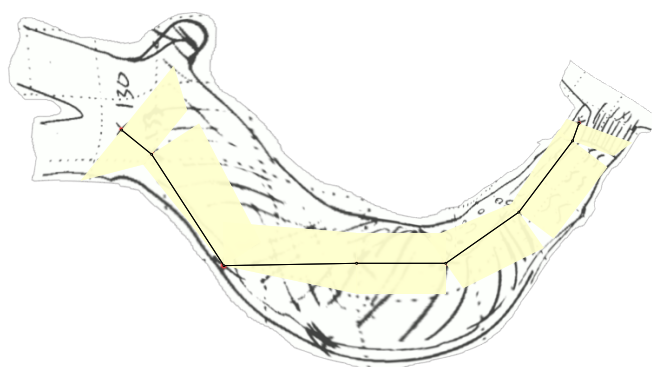
Стены после морфинга:



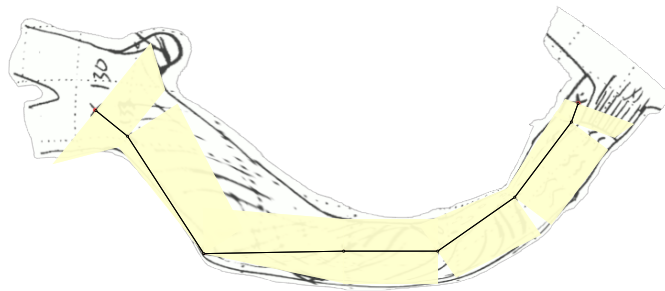
Другой эскиз съемки:



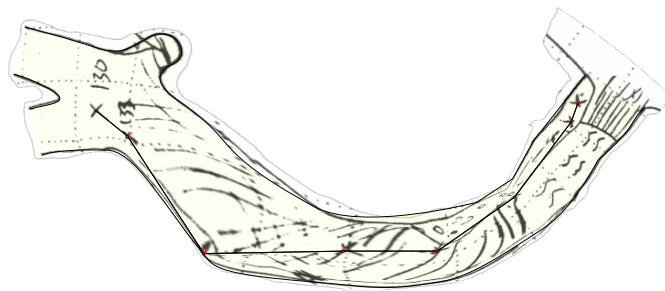
Это изображение после морфинга:



Эскиз подвергшийся морфингу после вставки точек *extra*:



И морфинг стен:

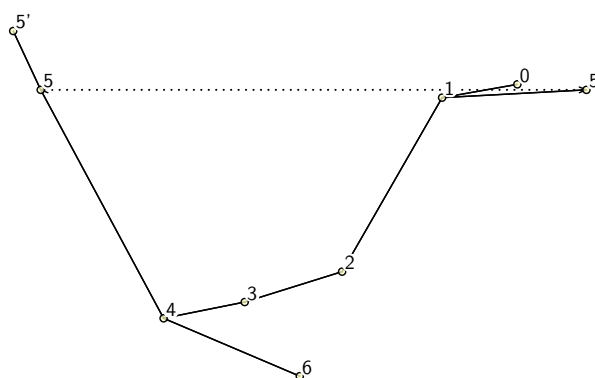


### Управление разрез-разверткой

Предположим следующую ситуацию (в плане) – есть кольцо нитки хода между станциями 1 и 4, небольшая труба возле станции 5 и вход в пещеру находится на станции 6.



По умолчанию разрез-развертка этой ситуации выглядит следующим образом:



Очевидно это не то, что мы хотели бы получить.

Чтобы контролировать процесс развертки существует специальная опция [extend](#) команды `centerline`.

Прежде всего, мы хотели бы начать нашу разрез-развертку на станции 6 (где находится вход). Это можно сделать, указав

`extend start 6`

в centerline. Теперь это выглядит лучше,



но по-прежнему существует проблема с ниткой, содержащей станцию 5.

Если мы хотим начать эту нитку со станции 4, нам необходимо запретить Therion'у разворот станции 5 из станции 1. Это можно сделать, указав

`extend ignore 1 5`

Это означает, что замер со станции 1 на станцию 5 будет проигнорирован, когда генерируется разрез-развертка.

Если мы хотим развернуть нитку начиная со станции 5 влево, нам нужно указать также

`extend left 5`

Это команда будет разворачивать все станции со станции 5 влево.



Как мы уже упоминали ранее, на станции 5 имеется небольшая труба. В этом случае гораздо более естественно сделать этот замер вертикальным (потому что это труба). Чтобы сделать это, используйте

extend vertical 5 5'



Или вы можете полностью скрыть этот замер, используя:

extend hide 5 5'



### Станции в скрапах разрез-развертки

Даже если станция 1 присутствует более одного раза на карте, Therion автоматически определяет правильное положение этой станции в каждом скрапе, и они обычно рисуются правильно.



Единственное, чего не хватает, это линия соединения между станциями 1 в этих двух нитках. Therion автоматически не генерирует эти линии, потому что их форма обычно зависит от конкретной карты.

Чтобы нарисовать такую линию, вам нужно всего лишь создать простой скрап с этой линией. Вот пример:

```
scrap sc -proj extended
  point 0 0 station -name 1 -from 5 -visibility off
  point 100 0 station -name 1 -from 2 -visibility off
  line map-connection
    0 5
    0 15
    100 15
    100 5
  endlines
endscrap
```

Как вы видите, даже если есть две станции с одинаковым именем, они отличаются друг от друга опцией **-from**, которая указывает предыдущую станцию в разрез-развертке. Используя этот скрап, вы получаете окончательную карту:





## Отображение перекрывающихся карт в смещении

Предположим, существует две карты  $m1$  и  $m2$  ( $m1$  выше  $m2$ ). По умолчанию Therion отображает  $m1$  поверх  $m2$ .



Чтобы отобразить карту  $m2$  в смещении, вам нужно создать карту, содержащую  $m2$ , и указать тип смещения и предварительного просмотра для  $m2$  на этой карте. Пример:

```
map m12
  m1
  break
  m2 [0 8 m] below
endmap
```

После выбора  $m12$  в файле конфигурации вы получите:



Соединительные стрелки создаются из точек *map-connection*, указанных в смещающихся скрапах. В противном случае эти точки игнорируются.

Если ситуация более сложная, например, существует  $m3$  ниже  $m2$



вам нужно создать более сложную структуру карты, чтобы отобразить это правильно.

```
map m23
  m2
  break
  m3 [-8 0 m] below
endmap
```

```
map m123
  m1
  break
  m23 [0 8 m] below
endmap
```

Тогда *m123* выглядит так:



### Импорт *survex* .3d файлов

Съемки Therion'a почти соответствуют префиксам станций, определяемым парой *\*begin/\*end* в файлах *survex*. Обычно между ними не существует взаимно однозначного отображения. Поэтому, если вы хотите сохранить

свою нитку хода указанную в файлах *survex*, вам нужно решить проблему, как совместить структуры данных *survex* и Therion'a. В этом примере показаны три разных способа, как иметь дело с префиксами станций при импорте файлов *survex* .3d в Therion.

### Использование съемки в .th файлах

Если вы импортируете файл .3d с *-surveys use*, то Therion пытается найти совпадение между *префиксом survex* и *названием съемки Therion'a*. Если это совпадение найдено, то станции вставляются в найденную съемку. В противном случае префикс остается с именами станций. Пример кода:

```
import use.3d -surveys use
input use-out.th2
survey use
  input use-in.th2
endsurvey
```

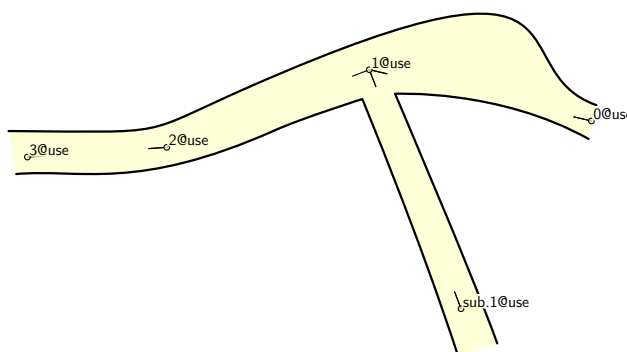
В этом случае вам следует позаботиться о том, где вы вводите ваши файлы .th2, содержащие скрапы. В файле *use-out.th2* вы должны ссылаться на имена станций, используя:

```
point 165.744 176.58 station -name 2@use
```

но в файле *use-in.th*, используя

```
point 321.454 236.22 station -name 1
```

Карта с названиями станций в этом случае выглядит так:



Названия станций перед символом '@', названия съемок после этого символа.

### Создание несуществующих съемок

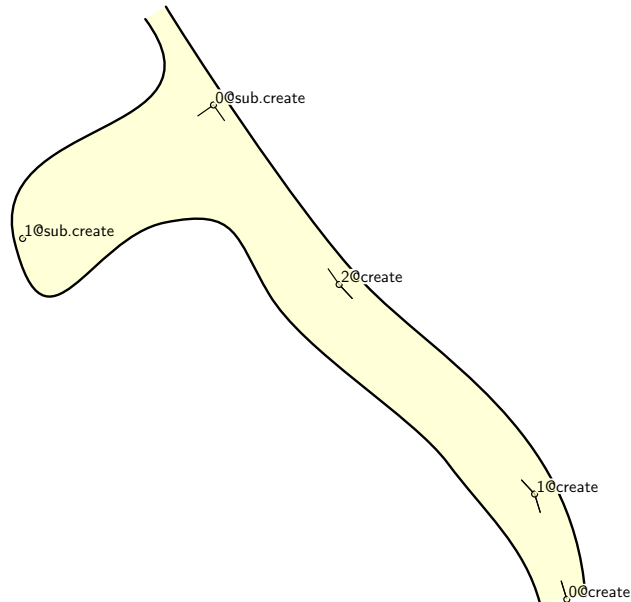
Если вы импортируете файл .3d, используя

```
import create.3d -surveys create
```

все префиксы *survex* обрабатываются, и если съемки с соответствующими именами не существуют, то они создаются. В этом случае, если вы сделаете

`input create.th2`

прямо после команды `import`, то вы ссылаетесь на станции в файле `.th2` с именами, указанными на карте.

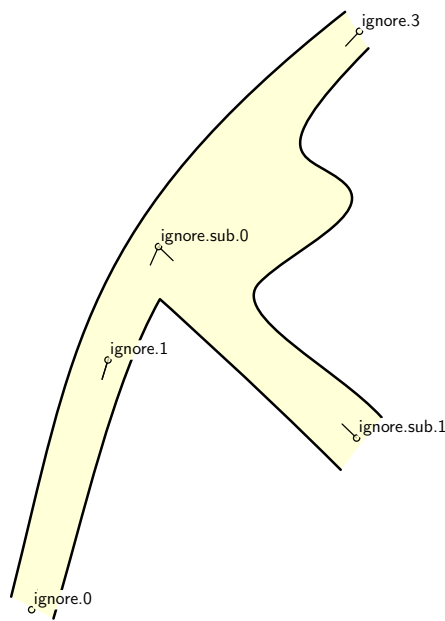


### Игнорирование префиксов станций

Последняя возможность – импортировать файл `.3d`, используя

`import ignore.3d -surveys ignore`

Также в этом случае ссылки станций в соответствующем файле `.th2` совпадают с именами станций на следующей карте.



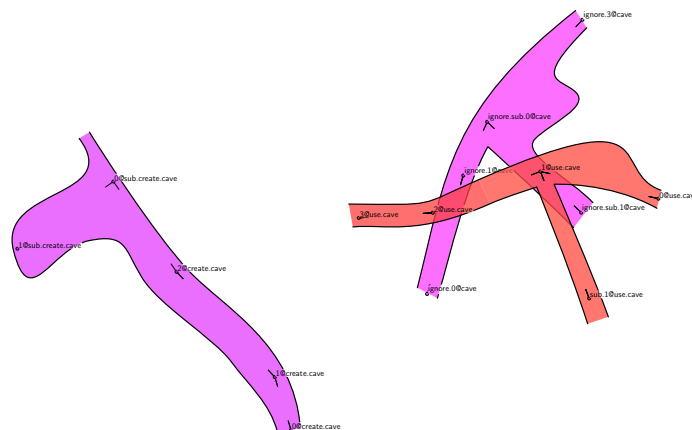
Обратите внимание, что в этом случае не создаются съемки. Названия станций берутся из .3d файлов без изменений.

### Управление большими проектами

Предположим ситуацию, когда вы хотите присоединиться к этим трем небольшим картам в рамках одного крупного проекта. Допустим, что координаты входных пещер указаны в файле верхнего уровня cave.3d. Если ваш код присоединения будет

```
import cave.3d -surveys use
survey cave
  input use/use.th
  input create/create.th
  input ignore/ignore.th
endsurvey
```

не все станции будут заменены правильно. "Created" серия помещается неправильно на карте.

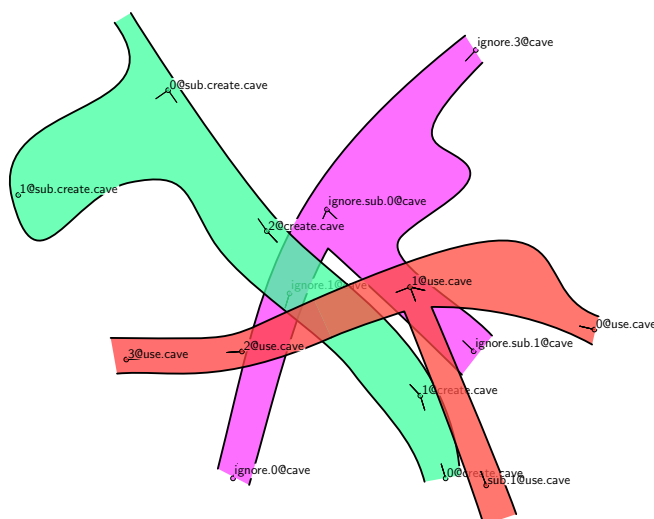


Это происходит потому, что в команде импорта верхнего уровня съемка *create* импортируется из файла *create/create.3d* с неправильными координатами, и мы импортируем файл *.3d* верхнего уровня с *-surveys use*. Это значит, что *cave.create.1* будет импортироваться здесь как *create.1@cave*, а не *1@create.cave*.

Чтобы решить эту проблему, нам нужно снова импортировать станции из файла *cave.3d* верхнего уровня с *-surveys create*.

```
import cave.3d -surveys use
import cave.3d -surveys create
survey cave
  input use/use.th
  input create/create.th
  input ignore/ignore.th
endsurvey
```

После этого дополнительного *import* окончательный вариант выглядит так, как ожидалось.



## Выводы

Даже если есть несколько возможностей, как сопоставить структуру префикса карты `survex` со структурой съемки Therion'a, самым чистым решением является создание структуры в `.th` файлах с использованием пустой пары `survey/endurvey` и всегда использовать `-surveys use` при импорте файлов `.3d`.

## Обработка вопросительных знаков

Возможные продолжения обрабатываются особым образом в области как нитки хода, так и в файлах карт. Вы можете связать текстовое описание, исследуемую длину (хода за этим знаком) или любой другой атрибут (например, `Code`, если у вас есть собственный стандарт кодирования для продолжения).

### Вопросительные знаки в нитке хода

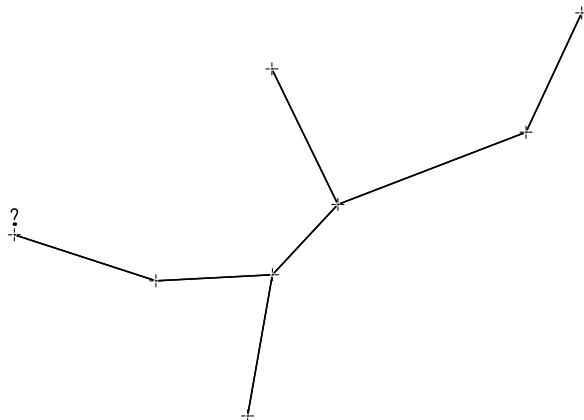
В объекте `centerline` вы можете добавить специальные флаги к станции, где возможно продолжение. Используйте следующий синтаксис

`station 5 "pit" continuation attr Code V explored 20m`

Когда вы экспортируете карту и используете

`symbol-show point flag:continuation`

в вашем `layout` станция с указанным символом продолжения помечена знаком вопроса (символ продолжения отображается над станцией).



Вы можете переопределить символ продолжения, чтобы показать также описание продолжения (сохраненное в атрибуте `_text`), используя следующий код в `layout`



```

code metapost
  def p_continuation(expr pos,theta,sc,al) =

% draw default continuation symbol
  p_continuation_UIS(pos,theta,sc,al);

% if text attribute is set
  if known(ATTR__text) and picture(ATTR__text):

% set labeling color to light orange
    push_label_fill_color(1.0, 0.9, 0.8);

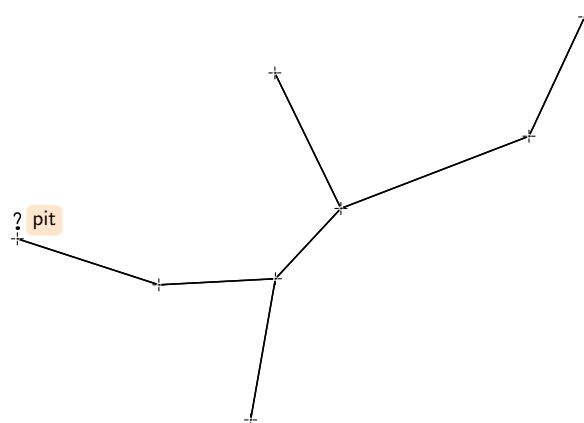
% draw filled label with text next to ?
    p_label.urt(ATTR__text,(.5u,-.25u) transformed T,0.0,8);

% restore original labeling color
    pop_label_fill_color;

  fi;
enddef;
endcode

```

Таким образом отображается описание продолжения на карте.



## Вопросительные знаки на картах

В скрапе (файл .th2) вы можете использовать опции `-text`, `-code` и `-explored` для символа продолжения.

```

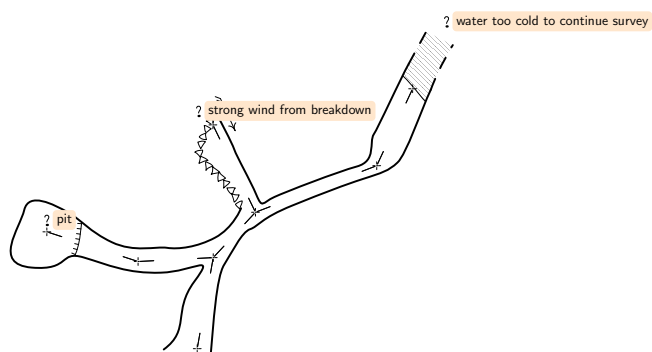
point 796.0 676.0 continuation -attr Code A -explored 50m \
  -text "слишком холодная вода для продолжения съемки"

```

На карте также отображаются только вопросительные знаки по умолчанию.



Когда вы переопределяете символ продолжения, как указано выше, вы также можете показать коды продолжения и описания.



### Экспортировать списки вопросительных знаков

Вы также можете экспортировать список всех продолжений из своего проекта, используя

[export continuation-list -o questions.html](#)

Файл questions.html содержит следующий список:

Комментарий	Исследовано	Съемка	Станция	Код
слишком холодная вода	50.0	Пример пещеры	7	A
сильная тяга из разлома		Пример пещеры	3	B
попор	20.0	Пример пещеры	5	V

## Использование пользовательских типов символов

Если Therion не содержит подходящий символ для ваших функций, вы можете нарисовать свой символ на карте, используя определенный тип символа (тип `u`, применяется для объектов точки, линии и области).

Синтаксис очень прост. Предположим, вы хотите создать точку, линию и область "bat". Вы просто используете `u:bat` как тип (bat – фактически подтип типа `u`). Таким образом, ваш код будет выглядеть так:

```
point 555.0 480.0 u:bat
```

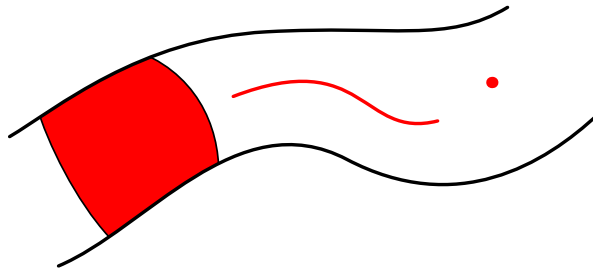
или

```
line u:bat
```

или

```
area u:bat
```

Когда вы экспортируете карту без определения символов в METAPOST, пользовательские символы выделяются красным цветом без какого-либо графического представления.



Чтобы правильно отображать их, вам необходимо определить символы для них в языке METAPOST, как обычные символы обычно переопределяются.

Первым переопределим символ точки. В

```
code metapost
```

в разделе вашего layout укажите символьную точку `u:bat`, подобный этому

```
def p_u_bat(expr pos, theta, sc, al) =  
  T := identity shifted pos;  
  thfill (bat_path scaled 2.0) withcolor black;  
enddef;
```

аналогично укажите в линейном символе `u:bat`

```
def l_u_bat(expr P) =  
  T:=identity;  
  cas := 0;  
  dlzka := arclength P;  
  mojkrok:=adjust_step(dlzka, 1.0u);  
  pickup PenD;  
  forever:  
    t := arctime cas of P;
```

```

        thfill bat_path scaled 0.5 shifted (point t of P) withcolor black;
        cas := cas + mojkrok;
        exitif cas > dlzka + (mojkrok / 3);% для ошибок округления
    endfor;
enddef;

```

и, наконец, область символа u:bat (шаблон в этом случае)

```

% bat pattern
beginpattern(pattern_bat);
    fill bat_path withcolor black;
endpattern;

% bat area symbol
def a_u_bat (expr Path) =
    T:=identity;
    thclean Path;
    thfill Path withpattern pattern_bat;
enddef;

```

Эти символы также будут включены в легенду. Чтобы изменить способ их рисования, просто определите соответствующий макрос. Его имя должно быть именем символьного макроса с суффиксом `_legend`.

```

def l_u_bat_legend =
    l_u_bat(((.2,.2) - (.8,.8)) inscale)
enddef;

```

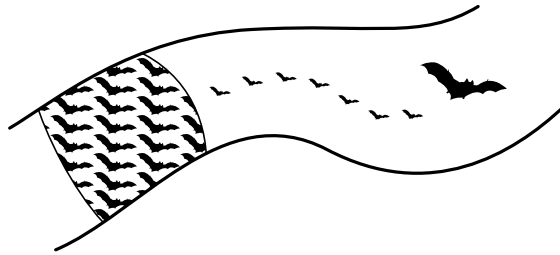
Finally, add description of your new symbols that will be shown in the legend using `text` command anywhere in the configuration file.

```

text en "point u:bat" "bat"
text en "line u:bat" "bat path"
text en "area u:bat" "lot of bats"

```

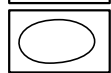
After all these definitions you receive bat point, line and area symbols with proper graphical representation and legend boxes.



# Legend



wall



border



bat



bat path



lot of bats