

The Therion Book

Stacho Mudrák

Martin Budaj



Therion is copyrighted software. Distributed under the GNU General Public License.

Copyright © 1999–2017 Stacho Mudrak, Martin Budaj

This book describes Therion 5.4.1 (2017-04-18).

Code contributions by *Olly Betts*, *Marco Corvi*, *Vladimir Georgiev*, *Georg Pacher* and *Dimitrios Zachariadis*.

We owe thanks to

Martin Sluka, *Ladislav Blazek*, *Martin Heller*, *Wookey*, *Olly Betts* and all users for their feedback, support and suggestions.

Переводы (%):

<i>Language</i>	<i>XTherion</i>	<i>Map header</i>	<i>Loch</i>	<i>Translated by</i>
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	–	Vasily V. Suhachev, Andrey Kozhenkov
sk	85	93	96	Stacho Mudrák
sq	85	87	–	Fatos Katallozi
zh	86	91	–	Zhang Yuan Hai, Duncan Collis

The cover picture shows survey sketch of *Hrozny kamenolom* Chamber in the Cave of Dead Bats in Slovakia and the map of it produced by Therion.

Содержание

Введение	5
Почему Therion?	5
Особенности	6
Требования к ПО	7
Инсталляция	8
Настройка среды	8
Как это работает?	8
Первый запуск	10
Создание файлов данных	11
Основы	11
Типы данных	12
Системы координат	13
Магнитное склонение	14
Формат данных	14
'encoding'	15
'input'	15
'survey'	15
'centreline'	17
'scrap'	21
'point'	24
'line'	27
'area'	30
'join'	30
'equate'	31
'map'	32
'surface'	33
'import'	34
'grade'	34
'revise'	35
Custom attributes	35
XTherion	35
XTherion – текстовый редактор	36
XTherion – редактор карт	36
Дополнительные инструменты	40
Сочитания клавишь и мыши в редакторе карт	41
Несколько мыслей о Therion'e	43
Как ввести нитку хода?	43
Как рисовать карты?	44
Как создавать модели?	45
Therion in depth	45
How the map is put together	45
Processing data	48
Configuration file	48
'system'	48
'encoding'	48
'language'	48
'cs'	48
'sketch-warp'	49
'input'	49

'source'	49
'select'	50
'unselect'	50
'text'	51
'layout'	51
'setup3d'	56
'sketch-colors'	56
'export'	57
Running Therion	59
XTherion—compiler	61
What we get?	62
Information files	62
Log file	62
XTherion	62
SQL export	62
Lists—caves, surveys, continuations	64
2D maps	64
Maps for printing	64
Maps for GIS	65
Special-purpose maps	65
3D models	65
Loch	65
Changing layout of PDF maps	66
Page layout in the atlas mode	66
Page layout in the map mode	71
Customizing text labels	72
New map symbols	72
Point symbols	73
Line symbols	74
Area symbols	74
Special symbols	75
Appendix	76
Compilation	76
Quick start	77
Hacker's guide	77
Environment variables	79
Initialization files	79
Therion	79
XTherion	82
Limitations	82
Example data	82
History	83
Future	85
General	85
2D maps	85
3D models	85
XTherion	85
Loch	85
Labyrinth	86

LET NO ONE IGNORANT OF GEOMETRY ENTER HERE
ΑΓΕΩΜΕΤΡΗΤΟΣ ΜΗΔΕΙΣ ΕΙΣΙΤΩ

—alleged inscription over the entrance
of Plato's Academy, 4th century BC

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, Survox — отличная программа для обработки нитки хода, и T_EX — 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 (Zurich). 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, 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, 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 читаемыми 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 *tkImg*. Это требуется только для XTherion.
- 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. Его необходимо скомпилировать и установить в соответствии с инструкциями в *Приложении*.

Установка в Windows:

Запустите программу установки и следуйте инструкциям. Он устанавливает все необходимые материалы и создает ярлыки для XTherion и Therion Book.

Настройка среды

Therion считывает настройки из файла инициализации. Настройки по умолчанию должны работать отлично для пользователей использующих только латинские символы¹, стандартные T_EX и METAPOST.

Если вы хотите использовать собственные шрифты для латинских или нелатинских 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:

¹ 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.



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), 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. Значение цветов:

- черный — программы и макросы Therion'a (XTherion написан на Tcl/Tk, поэтому для него требуется этот интерпретатор);
- красный — \TeX 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 pdf \TeX . Pdf \TeX 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 *Старт*). 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 T_EX. 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.

Only for you, children of doctrine and learning, have we written this work. Examine this book, ponder the meaning we have dispersed in various places and gathered again; what we have concealed in one place we have disclosed in another, that it may be understood by your wisdom.

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

—Henricus C. Agrippa ab Nettesheym, 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 [-option1 value1 -option2 value2 ...]`

где *arg1 ... argN* являются обязательными аргументами, а пары *-option value* являются параметрами, которые вы можете свободно пропустить. 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 [-option1 value1 -option2 value2 ...]`

`...`

`optionX valueX`

`data`

`...`

`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 одина 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 строки, 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 зазора.
- Все, что следует за # и до конца строки, даже внутри команды, считается комментарием и игнорируется.

5.4 • Многострочные комментарии внутри `comment ... endcomment` блока разрешены в файлах данных и конфигурационных файлах.

Типы данных

Therion использует следующие типы данных:

- *keyword* ▷ последовательность A-Z, a-z, 0-9 и _/- символов (не начинающиеся с '-').
- *ext_keyword* ▷ слово, которое также может содержать +*,.' символы, но не в первой позиции.
- *date* ▷ 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.
- *person* ▷ имя и фамилия человека, разделенные пробельными символами. Используйте '/' чтобы отделить имя и фамилию, если есть несколько имен.
- 5.3 • *string* ▷ 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.

- *units* ▷ поддерживаемые единицы длины: meter[s], centimeter[s], inch[es], feet[s], yard[s] (можно сокращать m, cm, in, ft, yd). Поддерживаемые угловые единицы: degree[s], minute[s] (можно сокращать deg, min), grad[s], mil[s], percent[age] (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:<number>` ▷ Большинство систем координат EPSG. Почти каждая система координат, используемая во всем мире, имеет собственный номер EPSG. Чтобы найти номер вашей системы, см. [extern/proj4/nad/epsg](#) файл в дистрибутиве исходников.
- `ESRI:<number>` ▷ 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.

5.4 • OSGB:<N, N, O, S или T><A-Z исключая I> ▷ Британская Национальная Сеть.

- S-MERC ▷ сферическая проекция Меркатора, используемая различными сайтами онлайн-сопоставления.

Магнитное склонение

5.4 Therion содержит встроенный IGRF² — модель геомагнитного поля Земли, действительная для периода 1900–2020 гг. Он автоматически используется, если пещера находится в пространстве с использованием любой из поддерживаемых геодезических систем координат, и никакое склонение не определяется пользователем. Вычисленное склонение указано в файле LOG для информации.

Формат данных

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

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

Описание: примечания относительно этой команды.

Синтаксис: описание синтаксиса.

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.

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

'encoding'

Описание:

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

Синтаксис:

`encoding <encoding-name>`

Контекст:

Это должна быть самая первая команда в файле.

Аргументы:

- `<encoding-name>` ▸ 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 портабельности используйте относительные пути, а для разделения каталогов используйте характерный для Unix '/', а не обратный слеш используемый в Windows '\\'.
Синтаксис:

Синтаксис:

`input <file-name>`

Контекст:

all

Аргументы:

- `<file-name>`

'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, указанное его `<id>` аргументом. Объекты (например станции или скрапы, см. ниже), которые относятся к subsurvey текущего survey, записываются как

`<object-id>@<subsurvey-id>`,

или, если есть больше уровней вложения

`<object-id>@<subsubsurvey-id>.<subsurvey-id>`.³

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 <id> [OPTIONS]`

... другие объекты ...

`endsurvey [<id>]`

Контекст:

none, survey

Аргументы:

- `<id>` ▷ идентификатор

Опции:

- `namespace <on/off>` ▷ указывает, создавать ли survey пространство имен (on по умолчанию).
- `declination <specification>` ▷ устанавливает склонение по умолчанию для всех объектов данных в этом survey (которые могут быть переопределены новым склонением в subsurveys). The `<specification>` имеет три формы:
 1. [] пустую строку. Это приведет к сбросу склонения.
 2. [`<значение>` `<единицы>`] установит одно значение (также для undated 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.

³ Примечание: невозможно связать любой объект с survey более высокого уровня.

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.

'centreline'

Описание:

Описание данных survey (нити 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` [OPTIONS]

```
date <дата>
team <персона> [<роли>]
explo-date <дата>
explo-team <персона>
instrument <quantity list> <описание>
calibrate <quantity list> <zero error> [<scale>]
units <quantity list> [<factor>] <единицы>
sd <quantity list> <значение> <единицы>
grade <grade list>
declination <значение> <единицы>
grid-angle <значение> <единицы>
infer <what> <on/off>
mark <тип>
flags <shot flags>
station <станция> <комментарий> [<flags>]
cs <система координат>
fix <станция> [<x> <y> <z> [<std x> <std y> <std z>]]
equate <station list>
```

```

data <стиль> <readings order>
break
group
endgroup
walls <auto/on/off>
vthreshold <число> <единицы>
extend <список> [<станция> [<станция>]]
station-names <префикс> <суффикс>
...
[данные съемки]
...
endcentreline

```

Контекст:

none, survey

Опции:

- `id <ext_keyword>` ▷ id 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 значения: station, length, tape, [back]compass, [back]bearing, [back]clino, [back]gradient, counter, depth, station, position, notes, pictures, pics, instruments (insts), assistant (dog).
- `explo-team <персона>` ▷ член исследовательского отделения.
- `instrument <quantity list> <description>` ▷ описание инструмента, который использовался для определения количественных данных (те же значения, что и роль человека в команде)
- `infer <what> <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' will case program to

interpret shots with 0 length as equate commands (which means that no tape corrections are applied)

- `declination <value> <units>` ▸ sets the declination for subsequent shots

$$\text{true bearing} = \text{measured bearing} + \text{declination}.$$

The declination is positive when the magnetic north is east of true north. If no declination is specified, or the declination is reset (-), then a valid declination specification is searched for in all surveys the data object is in. См. опцию `declination` команды `survey`.

- `grid-angle <value> <units>` ▸ specifies the magnetic grid angle (declination against grid north).
- `sd <quantity list> <значение> <единицы>` ▸ задает стандартное отклонение для данных измерений. Quantity list 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>` ▸ sets standard deviations according to the survey grade specification (see grade command). All previously specified standard deviations or grades are lost. If you want to change an SD, use the sd option after this command. If multiple grades are specified, only the last one applies. You can specify grades only for position or only for surveys. If you want to combine them, you must use them in one grade line.
- `units <quantity list> [<factor>] <units>` ▸ set the units for given measurements (same quantities as for sd).
- `calibrate <quantity list> <zero error> [<scale>]` ▸ set the instrument calibration. The measured value is calculated using the following formula: $\text{measured value} = (\text{read value} - \text{zero error}) \times \text{scale}$. The supported quantities are the same as sd.
- `break` ▸ can be used with interleaved data to separate two traverses
- `mark [<station list>] <type>` ▸ set the type of named stations. `<type>` is one of: fixed, painted and temporary (default). If there is no station list, all subsequent stations are marked.
- `flags <shot flags>` ▸ set flags for following shots. The supported flags are: `surface` (for surface measurements), `duplicate` (for duplicated surveys), `splay` (for short side legs that are hidden in maps and models by default). These are excluded from length calculations.

All shots having one of the stations named either '.' or '-' are `splay` shots by default (see also `data` command).

If flag is set to `approx[imate]`, it is included to total length calculations, but also displayed separately in survey statistics. It should be used for shots, that were not surveyed properly and need to be resurveyed.

Also “not” is allowed before a flag.

- `station <station> <comment> [<flags>]` ▷ set the station comment and its flags. If "" is specified as a comment, it is ignored.

5.3 Supported flags: `entrance`, `continuation`, `air-draught[:winter/summer]`, `sink`, `spring`, `doline`, `dig`, `arch`, `overhang`. Also `not` is allowed before a flag, to remove previously added flag.

You can also specify custom attributes to the station using `attr` flag followed by attribute name and value. Example:

```
station 4 "pit to explore" continuation attr code "V"
```

If there is a passage, that was explored, but not surveyed yet, estimated explored length of this passage can be added to the station with `continuation` flag. Just add `explored <explored-length>` to the station flags. Explored lengths are a part of survey/cave statistics, displayed separately. Example:

```
station 40 "ugly crollway" continuation explored 100m
```

- `cs <coordinate system>` ▷ coordinate system for stations with fixed coordinates
- `fix <station> [<x> <y> <z> [<std x> <std y> <std z>]]` ▷ fix station coordinates (with specified errors—only the units transformation, not calibration, is applied to them).
- `equate <station list>` ▷ set points that are equivalent
- `data <style> <readings order>` ▷ set data style (normal, toprofil, diving, cartesian, cylpolar, dimensions, nosurvey) and readings order. Reading is one of the following keywords: `station`, `from`, `to`, `tape/length`, `[back]compass/[back]bearing`, `[back]clino/[back]gradient`, `depth`, `fromdepth`, `todepth`, `depthchange`, `counter`, `fromcount`, `tocount`, `northing`, `easting`, `altitude`, `up/ceiling`⁴, `down/floor`, `left`, `right`, `ignore`, `ignoreall`.

See Survex manual for details.

For interleaved data both newline and direction keywords are supported. If backward and forward compass or clino reading are given, the average of them is computed.

5.3 If one of the shot stations is named either `'.'` or `'-'`, the shot has `splay` attribute set. *Dot* should be used for shots ending on features inside passage, *dash* for shots ending on passage walls, floor or ceiling. Although Therion makes no distinction between them yet, it should be used to improve 3D modeling in the future.

- `group`

⁴ dimension may be specified as a pair `[<from> <to>]`, meaning the size at the beginning and end of the shot

- `endgroup` ▷ `group/endgroup` pair enables the user to make temporary changes in almost any setting (calibrate, units, sd, data, flags...)
- `walls <auto/on/off>` ▷ turn on/off passage shape generation from LRUD data for subsequent shots. If set `auto`, passage is generated only if there is no scrap referencing given centreline.
- `vthreshold <number> <units>` ▷ threshold for interpreting LRUD readings as left-right-front-back reading perpendicular to the shot.

If passages are horizontal (`inclination < vthreshold`), LR is perpendicular to the shot and UD is vertical.

If passages are more or less vertical (`inclination > vthreshold`), even UD becomes perpendicular to the shot – otherwise passages would not look very good. In the case of vertical shots, UD is interpreted as north-south dimension from the station to allow tube-like modelling of verticals.

- `extend <spec> [<station> [<station>]]` ▷ control how the centerline is extended. `<spec>` is one of the following

`normal/reverse` ▷ extend given and following stations to the same/reverse direction as previous station. If two stations are given—direction is applied only to given shot.

`left/right` ▷ same as above, but direction is specified explicitly.

`vertical` ▷ do not move station (shot) in *X* direction, use only *Z* component of the shot

`start` ▷ specify starting station (shot)

`ignore` ▷ ignore specified station (shot), continue extended elevation with other station (shot) if possible

`hide` ▷ do not show specified station (shot) in extended elevation

If no stations are specified, `<spec>` is valid for following shots specified.

- `station-names <prefix> <suffix>` ▷ adds given prefix/suffix to all survey stations in the current centreline. Saves some typing.

'scrap'

Описание:

Scrap is a piece of 2D map, which doesn't contain overlapping passages (i.e. all the passages may be drawn on the paper without overlapping). For small and simple caves, the whole cave may belong to one scrap. In complicated systems, a scrap is usually one chamber or one passage. Ideally, a scrap contains about

100 m of the cave.⁵ Each scrap is processed separately by METAPOST; scraps which are too large may exceed METAPOST's memory and cause errors.

Scrap consists of point, line and area map symbols. See chapter *How the map is put together* for explanation how and in which order are they displayed.

Scrap border consists of lines with the `-outline out` or `-outline in` options (passage walls have `-outline out` by default). These lines shouldn't intersect—otherwise Therion (METAPOST) can't determine the interior of the scrap and METAPOST issues a warning message "`scrap outline intersects itself`".

Each scrap has its own local cartesian coordinate system, which usually corresponds with the millimeter paper (if you measure the coordinates of map symbols by hand) or pixels of the scanned image (if you use XTherion). Therion does the transformation from this local coordinate system to the real coordinates using the positions of survey stations, which are specified both in the scrap as point map symbols and in centreline data. If the scrap doesn't contain at least two survey stations with the `-name` reference, you have to use the `-scale` option for calibrating the scrap. (Это обычно используется для сечений.)

The transformation consists of the following steps:

- Linear transformation (shifting, scaling and rotation) which 'best' fits stations drawn in the scrap to real ones. 'Best' means that the sum of squared distances between corresponding stations before and after transformation is minimal. The result is displayed red if `debug` option of the `layout` command is set `on`.
- Non-linear transformation of the scrap which (1) moves survey stations to their correct position, (2) is continuous. Displayed blue in the `debug` mode.
- Non-linear transformation of the scrap which (1) moves joined points together, (2) doesn't move survey stations, (3) is continuous. Finally the position of curves' control points is adjusted to preserve smoothness. The result is final map.

Синтаксис: `scrap <id> [OPTIONS] ... point, line and area commands ... endscrap [<id>]`

Контекст:

none, survey

Аргументы:

- `<id>` ▸ scrap identifier

⁵ If necessary, scraps may be much smaller—just to display a few meters of the cave. *When deciding about scrap size please take into account the following:* Using small scraps may take more time for cartographer to optimize scrap joins. On the other hand smaller scraps will probably be less distorted by map warping algorithms than larger scraps. Using too large scraps may exhaust METAPOST's memory if passage fills are frequently used and the map editor in XTherion is much less responsive when editing huge scraps.

Onquuu:

- **projection** *<specification>* ▷ specifies the drawing projection. Each projection is identified by a type and optionally by an index in the form *type[:index]*. The index can be any keyword. The following projection types are supported:
 1. **none** ▷ no projection, used for cross sections or maps that are independent of survey data (e.g. digitization of old maps where no centreline data are available). No index is allowed for this projection.
 2. **plan** ▷ basic plan projection (default).
 3. **elevation** ▷ orthogonal projection (a.k.a. projected profile) which optionally takes a view direction as an argument (e.g. *[elevation 10]* or *[elevation 10 deg]*).
 4. **extended** ▷ extended elevation (a.k.a. extended profile).
- **scale** *<specification>* ▷ is used to pre-scale (convert coordinates from pixels to meters) the scrap data. If scrap projection is none, this is the only transformation that is done with coordinates. The *<specification>* has four forms:
 1. *<number>* ▷ *<number>* meters per drawing unit.
 2. *[<number> <length units>]* ▷ *<number> <length units>* per drawing unit.
 3. *[<num1> <num2> <length units>]* ▷ *<num1>* drawing units corresponds to *<num2> <length units>* in reality.
 4. *[<num1> ... <num8> [<length units>]]* ▷ this is the most general format, where you specify, in order, the *x* and *y* coordinates of two points in the scrap and two points in reality. Optionally, you can also specify units for the coordinates of the 'points in reality'. This form allows you to apply both scaling and rotation to the scrap.
- **cs** *<coordinate system>* ▷ assumes that (calibrated) local scrap coordinates are given in specified coordinate system. It is useful for absolute placing of imported sketches where no survey stations are specified.⁶
- **stations** *<list of station names>* ▷ stations you want to plot to the scrap, but which are not used for scrap transformation. You don't have to specify (draw) them with the **point station** command.
- **sketch** *<filename> <x> <y>* ▷ underlying sketch bitmap specification (lower left corner coordinates).
- **walls** *<on/off/auto>* ▷ specify if the scrap should be used in 3D model reconstruction
- **flip** *(none)/horizontal/vertical* ▷ flips the scrap after scale transformation

⁶ If there are some survey stations in the scrap, the **cs** specification is ignored.

- `station-names <prefix> <suffix>` ▸ adds given prefix/suffix to all survey stations in the current scrap. Saves some typing.
- `author <date> <person>` ▸ author of the data and its creation date
- `copyright <date> <string>` ▸ copyright date and name
- `title <string>` ▸ description of the object

'point'

Описание:

Point is a command for drawing a point map symbol.

Синтаксис:

`point <x> <y> <type> [OPTIONS]`

Контекст:

scrap

Аргументы:

- `<x>` and `<y>` are the drawing coordinates of an object.
- `<type>` determines the type of an object. The following types are supported:

special objects: `station`⁷, `section`⁸, `dimensions`⁹;

labels: `label`, `remark`, `altitude`¹⁰, `height`¹¹, `passage-height`¹², `station-name`¹³, `date`;

*symbolic passage fills:*¹⁴ `bedrock`, `sand`, `raft`, `clay`, `pebbles`, `debris`, `blocks`, `water`, `ice`, `guano`, `snow`;

speleothems: `flowstone`, `moonmilk`, `stalactite`, `stalagmite`, `pillar`, `curtain`, `helictite`, `soda-straw`, `crystal`, `wall-calcite`, `popcorn`, `disk`, `gypsum`, `gypsum-flower`, `aragonite`, `cave-pearl`, `rimstone-pool`, `rimstone-dam`, `anastomosis`, `karren`, `scallop`, `flute`, `raft-cone`, `clay-tree`;

5.4

⁷ Survey station. For each scrap (with the exception of scraps in 'none' projection) at least one station with station reference (`-name` option) has to be specified.

⁸ `section` is an anchor for placing the cross-section at this point. This symbol has no visual representation. The cross section must be in the separate scrap with 'none' projection specified. You can specify it through the `-scrap` option.

⁹ Use `-value` option to specify passage dimensions above/below centerline plane used while creating 3D model.

¹⁰ General altitude label. All altitudes are exported as a difference against grid Z origin (which is 0 by default). To display altitude on the passage wall, use `altitude` option for any line point of the passage wall

¹¹ Height of formations inside of the passage (like pit etc.); see below for details.

¹² Height of the passage; see below for details.

¹³ If no text is specified, the name of the nearest station is used.

¹⁴ Unlike other point symbols, these are clipped by the scrap border. See the chapter *How the map is put together*.

equipment: anchor, rope, fixed-ladder, rope-ladder, steps, bridge, traverse, camp, no-equipment;

passage ends: continuation, narrow-end, low-end, flowstone-choke, breakdown-choke, clay-choke, entrance; 5.4

others: dig, archeo-material, paleo-material, vegetable-debris, root, water-flow, spring¹⁵, sink, ice-stalactite, ice-stalagmite, ice-pillar, gradient, air-draught¹⁶, map-connection¹⁷, extra¹⁸, u¹⁹. 5.4

Onquu:

- **subtype** <keyword> ▷ determines the object's subtype. The following subtypes for given types are supported:

*station:*²⁰ temporary (default), painted, natural, fixed;

air-draught: winter, summer, undefined (default);

water-flow: permanent (default), intermittent, paleo.

The subtype may be specified also directly in <type> specification using ':' as a separator.²¹

Any subtype specification can be used with user defined type (u). In this case you need also to define corresponding metapost symbol (see the chapter *New map symbols*).

- **orientation/orient** <number> ▷ defines the orientation of the symbol. If not specified, it's oriented to north. $0 \leq \text{number} < 360$.
- **align** ▷ alignment of the symbol or text. The following values are accepted: center, c, top, t, bottom, b, left, l, right, r, top-left, tl, top-right, tr, bottom-left, bl, bottom-right, br.
- **scale** ▷ symbol scale, can be: tiny (xs), small (s), normal (m), large (l), huge (xl) or a numeric value. Normal is default. Named sizes scale by $\sqrt{2}$, so that $xs \equiv 0.5$, $s \equiv 0.707$, $m \equiv 1.0$, $l \equiv 1.414$ and $xl \equiv 2.0$.
- **place** <bottom/default/top> ▷ changes displaying order in the map.
- **clip** <on/off> ▷ specify whether a symbol is clipped by the scrap border. You cannot specify this option for the following symbols: station, station-name, label, remark, date, altitude, height, passage-height.

¹⁵ Always use **spring** and **sink** symbols with a **water-flow** arrow.

¹⁶ Number of ticks is set according to **-scale** option

¹⁷ Virtual point, used to indicate connection between shifted maps (extended elevation, map offset).

¹⁸ Additional morphing point.

¹⁹ For user defined point symbols.

²⁰ if station subtype is not specified, Therion reads it from centreline, if it's specified there

²¹ E.g. **station:fixed**

- `dist <distance>` ▷ valid for extra points, specifies the distance to the nearest station (or station specified using `-from` option. If not specified, appropriate value from LRUD data is used.
- `from <station>` ▷ valid for extra points, specifies reference station.
- `visibility <on/off>` ▷ displays/hides the symbol.
- `context <point/line/area> <symbol-type>` ▷ (to be used with `symbol-hide` and `symbol-show` layout options) symbol will be hidden/shown according to rules for specified `<symbol-type>`.²²
- `id <ext_keyword>` ▷ ID of the symbol.

Type-specific options:

- `name <reference>` ▷ if the point type is station, this option gives the reference to the real survey station.
- `extend [prev[ious] <station>]` ▷ if the point type is station and scrap projection is extended elevation, you can adjust the extension of the centreline using this option.
- `scrap <reference>` ▷ if the point type is section, this is a reference to a cross-section scrap.
- `explored <length>` ▷ if the point type is continuation, you can specify length of passages explored but not surveyed yet. This value is afterwards displayed in survey/cave statistics.
- `text` ▷ text of the label, remark or continuation. It may contain following formatting keywords:²³

`
` ▷ line break

`<center>/<centre>`, `<left>`, `<right>` ▷ line alignment for multi-line labels. Ignored if there is no `
` tag.

`<thsp>` ▷ thin space

`<rm>`, `<it>`, `<bf>`, `<ss>`, `<si>` ▷ font switches

5.3 `<rtl>` and `</rtl>` ▷ marks beginning and end of a right-to-left written text

5.3 `<lang:XX>` ▷ creates multilingual label (see `string` type for detailed description)

- `value` ▷ value of height, passage-height or altitude label or point dimensions
height: according to the sign of the value (positive, negative or unsigned), this type of symbol represents chimney height, pit depth or step height in general.

²² Example: if you specify `-context point air-draught` to a label which displays the observation date, the `symbol-hide point air-draught` command would hide both air-draught arrow and the corresponding label.

²³ For SVG output, only `
`, `<thsp>`, `<it>`, `<bf>`, `<rm>` and `<lang:XX>` keywords are taken into account; all others are silently ignored.

The numeric value can be optionally followed by '?', if the value is presumed and units can be added (e.g. `-value [40? ft]`).

passage-height: the following four forms of value are supported: `+<number>` (the height of the ceiling), `-<number>` (the depth of the floor or water depth), `<number>` (the distance between floor and ceiling) and `[+<number> -<number>]` (the distance to ceiling and distance to floor).

altitude: the value specified is the altitude difference from the nearest station. If the altitude value is prefixed by "fix" (e.g. `-value [fix 1300]`), this value is used as an absolute altitude. The value can optionally be followed by length units.

dimensions: `-value [<above> <below> [<units>]]` specifies passage dimensions above/below centerline plane used in 3D model.

'line'

Описание:

Line is a command for drawing a line symbol on the map. Each line symbol is oriented and its visualization may depend on its orientation (e.g. pitch edge ticks). The general rule is that the free space is on the left, rock on the right. Examples: the lower side of a pitch, higher side of a chimney and interior of a passage are on the left side of pitch, chimney or wall symbols, respectively.

Синтаксис:

`line <type> [OPTIONS] [OPTIONS] ... [LINE DATA] ... [OPTIONS] ... [LINE DATA] ... endl`

Контекст:

scrap

Аргументы:

- `<type>` is a keyword that determines the type of line. The following types are supported:

passages: `wall`, `contour`, `slope`²⁴, `floor-step`, `pit`, `ceiling-step`, `chimney`, `overhang`, `ceiling-meander`, `floor-meander`;

passage fills: `flowstone`, `moonmilk`, `rock-border`²⁵, `rock-edge`²⁶, `water-flow`;

labels: `label`;

²⁴ Slope line marks upper border of the sloping area. It's necessary to specify `l-size` in at least one point. Gradient lines length and orientation is an average of specified `l-sizes` and `orientations` in the nearest points. If there is no orientation specification, gradient marks are perpendicular to the slope line.

²⁵ Outer outline of large boulders. If the line is closed, it is filled with the background colour.

²⁶ Inner edges of large boulders.

special: `border`, `arrow`, `section`²⁷, `survey`²⁸, `map-connection`²⁹, `u`³⁰.

Опции командной строки:

- `subtype <keyword>` ▷ determines line subtype. The following subtypes are supported for given types:

5.4 *wall:* `invisible`, `bedrock` (default), `sand`, `clay`, `pebbles`, `debris`, `blocks`, `ice`, `underlying`, `overlying`, `unsurveyed`, `presumed`, `pit`³¹, `flowstone`, `moonmilk`;

border: `visible` (default), `invisible`, `temporary`, `presumed`;

water-flow: `permanent` (default), `conjectural`, `intermittent`;

survey: `cave` (default), `surface` (default if centreline has surface flag).

The subtype may be specified also directly in `<type>` specification using ':' as a separator.³²

Any subtype specification can be used with user defined type (`u`). In this case you need also to define corresponding metapost symbol (see the chapter *New map symbols*).

- `[LINE DATA]` specify either the coordinates of a line segment `<x> <y>`, or coordinates of a Bezier curve arc `<c1x> <c1y> <c2x> <c2y> <x> <y>`, where `c` indicates the control point.
- `close <on/off/auto>` ▷ determines whether a line is closed or not
- `mark <keyword>` ▷ is used to mark the point on the line (see `join` command).
- `orientation/orient <number>` ▷ orientation of the symbols on the line. If not specified, it's perpendicular to the line on its left side. $0 \leq \text{number} < 360$.
- `outline <in/out/none>` ▷ determines whether the line serves as a border line for a scrap. Default value is '`out`' for walls, '`none`' for all other lines. Use `-outline in` for large pillars etc.
- `reverse <on/off>` ▷ whether points are given in reverse order.
- `size <number>` ▷ line width (left and right sizes are set to one half of this value)
- `r-size <number>` ▷ size of the line to the right

²⁷ Line showing cross-section position. If both control points (red dots) of a Bezier curve (grey line) are given then the section line (blue) is drawn up to the perpendicular projection (dotted) of the first control point and from the projection (dotted) of the section control point. No section curve is displayed.



²⁸ Survey line is automatically drawn by Therion.

²⁹ Used to indicate connection between maps (in offset, or the same points in extended elevation).

³⁰ For user defined line symbols.

³¹ Usually open to surface.

³² E.g. `border:invisible`

- `l-size <number>` ▷ same to the left. Required for `slope` type.
- `smooth <on/off/auto>` ▷ whether the line is smooth at the given point. Auto is default.
- `adjust <horizontal/vertical>` ▷ shifts the line point to be aligned horizontally/vertically with the previous point (or next point if there is no previous point). The result is horizontal/vertical line segment). If all line points have this option, they are aligned to the average y or x coordinate, respectively. This option is not allowed in the `plan` projection.
- `place <bottom/default/top>` ▷ changes displaying order in the map.
- `clip <on/off>` ▷ specify whether a symbol is clipped by the scrap border.
- `visibility <on/off>` ▷ displays/hides the symbol.
- `context <point/line/area> <symbol-type>` ▷ (to be used with `symbol-hide` and `symbol-show` layout options) symbol will be hidden/shown according to rules for specified `<symbol-type>`.

Type-specific options:

- `altitude <value>` ▷ can be specified only with the wall type. This option creates an altitude label on the wall. All altitudes are exported as a difference against grid Z origin (which is 0 by default). If the value is specified, it gives the altitude difference of the point on the wall relative to the nearest station. The value can be prefixed by a keyword "`fix`", then no nearest station is taken into consideration; the absolute given value is used instead. Units can follow the value. Examples: `+4`, `[+4 m]`, `[fix 1510 m]`.
- `border <on/off>` ▷ this option can be specified only with the 'slope' symbol type. It switches on/off the border line of the slope.
- `direction <begin/end/both/none/point>` ▷ can be used only with the section type. It indicates where to put a direction arrow on the section line. None is default.
- `gradient <none/center/point>` ▷ can be used only with the contour type and indicates where to put a gradient mark on the contour line. If there is no gradient specification, behaviour is symbol-set dependent (e.g. no tick in UIS, tick in the middle in SKBB).
- `head <begin/end/both/none>` ▷ can be used only with the arrow type and indicates where to put an arrow head. End is default.
- `text <string>` ▷ valid only for label lines.
- `height <value>` ▷ height of pit or wall:pit; available in METAPOST as a numeric variable `ATTR__height`.

Onquu:

- `id <ext_keyword>` ▷ ID of the symbol.

'area'

Описание:

Area is specified by surrounding border lines. They may be of any type, but must be listed in order and each pair of consecutive lines must intersect. In order to be sure that lines intersect even after scrap transformation you may e.g. continue a lake border 1 cm behind a passage wall—these overlaps will be automatically clipped by scrap border. You may use invisible border to achieve this inside of the passage.

Синтаксис:

```
area <type> place <bottom/default/top> clip <on/off> visibility <on/off> ... border line refer-  
ences ... endarea
```

Контекст:

scrap

Аргументы:

- **<type>** is one of following: [water](#), [sump](#), [sand](#), [debris](#), [blocks](#), [flowstone](#), [moonmilk](#), [snow](#), [ice](#), [clay](#), [pebbles](#), [bedrock](#)³³, [u](#)³⁴.

Опции командной строки:

- the data lines consist of border line references (IDs)
- **place <bottom/default/top>** ▸ changes displaying order in the map.
- **clip <on/off>** ▸ specify whether a symbol is clipped by the scrap border.
- **visibility <on/off>** ▸ displays/hides the symbol.
- **context <point/line/area> <symbol-type>** ▸ (to be used with [symbol-hide](#) and [symbol-show](#) layout options) symbol will be hidden/shown according to rules for specified **<symbol-type>**.

Опции:

- **id <ext_keyword>** ▸ ID of the symbol.

'join'

Описание:

Join works in two modes: it joins either two scraps or two or more points or lines in a map together.

³³ An empty area which can be used to clean the background.

³⁴ For user defined area symbols, may be followed by arbitrary subtype.

When joining more than two points or lines, use one join command for all of them, not a sequence of join commands for pairs.³⁵

When joining scraps, only passage walls are joined. It's a good idea to place a scrap join in the passage which is as simple as possible, otherwise you have to specify join for each pair of objects which should be joined.³⁶

Синтаксис:

`join <point1> <point2> ... <pointN> [OPTIONS]`

Контекст:

none, scrap, survey

Аргументы:

- `<pointX>` can be an ID of a point or line symbol, optionally followed by a line point mark `<id>:<mark>` (e.g. `podangl_l31@podangl:mark1`). `<mark>` can be also 'end' (end of the line) or line point index (where 0 is the first point).

A special case is when `<point1>` and `<point2>` are scrap IDs—than the closest scrap ends are joined together.

Опции:

- `smooth <on/off>` indicates whether two lines are to be connected smoothly.
- `count <N>` (when used with scraps) ▸ Therion will try to join scraps which connect in `N` locations/passages.

'equate'

Описание:

Устанавливает эквивалентность станций съёмки.

Синтаксис:

`equate <список станций>`

Контекст:

none, survey

³⁵ E.g. use `join a b c`, not `join a b` followed by `join b c`.

³⁶ If you want some object which is clipped by a scrap boundary to continue to a neighbouring scrap, use `-clip off` option for that object.

'map'

Описание:

A map is a collection of either scraps or other maps of the same projection type. It's possible to include survey in the map—this will display centreline in the map. Map object simplifies the data management when selecting data for output. See the chapter *How the map is put together* for more thorough explanation.

Синтаксис:

```
map <id> [OPTIONS] ... scrap, survey or other map references ... break ... next level scrap, survey  
or other map references ... preview <above/below> <other map id> endmap
```

Контекст:

none, survey

Аргументы:

- <id> ▸ scrap identifier

Опции командной строки:

- the data lines consist of scrap or map references. Note that you can not mix them together.
- if you refer to map, you can specify offset at which this sub-map will be displayed together with preview type of its original position. Syntax is following:
<map reference> [<offset X> <offset Y> <units>] <above/below/none>
- scraps following the `break` will be placed on another level
- `preview <above/below> <other map id>` will put the outline of the other map in the specified preview position relative to the current map.

Preview is displayed only if the map is in the `map-level` level as specified by the `select` command.

Use the `revise` command if you want to add maps from higher levels to the preview.

- `colo[u]r <color>` ▸ set the map colour; this option overrides the automatic choice when the layout specifies `colour map-fg [map]`.

Опции:

- `projection/proj <plan/elevation/extended/none>` ▸ required if the map contains survey.
- `title <string>` ▸ description of the object
- `survey <id>` ▸ associate a survey with map (e.g. all surveying statistics from this survey will be used when this map is selected for output).

5.4

'surface'

Описание:

Surface (terrain) specification. It is possible to display it in two ways: as a scanned topographical map (both in 2D map and 3D model³⁷) or surface grid – digital elevation model (in 3D model only).

Синтаксис:

```
surface [<name>] cs <coordinate system> bitmap <filename> <calibration> grid-units <units>  
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 <coordinate system>` ▷ coordinate system for bitmap calibration and grid origin specification
- `bitmap <filename> <calibration>` ▷ scanned topographical map.

`calibration` may have two forms:

1. `[X1 Y1 x1 y1 X2 Y2 x2 y2 [units]]`, where upper case X/Y variables are picture coordinates (pixels; lower-left corner is 0 0), lower-case x/y variables are real coordinates. Optional units apply to real coordinates (metres by default).
2. `[X1 Y1 station1 X2 Y2 station2]`, where upper case X/Y variables are picture coordinates and `station1` and `station2` are survey stations names.

- `grid-units <units>` ▷ units in which grid is specified. Metres by default.
- `grid <origin x> <origin y> <x spacing> <y spacing> <x count> <y count>`
`<origin x> <origin y>` ▷ specify coordinates of the lower-left (S-W) corner of the grid
`<x spacing> <y spacing>` ▷ distance between grid nodes in W-E and S-N directions
`<x count> <y count>` ▷ number of nodes in the row and number of rows which form the grid (see below).
- `[grid data]` ▷ a stream of numbers giving the altitude a.s.l. in grid nodes. It starts in the grid-origin and fills the grid in rows (in the row from W to E; rows from S to N).
- `grid-flip (none)/vertical/horizontal` ▷ useful if your grid (exported from other program) needs to be flipped

³⁷ You need to enter elevation data in order to display the topographical map in 3D model. Currently only JPEG maps are supported in 3D.

'import'

Описание:

Reads survey data in different formats (currently processed centreline in *.3d, *.plt, *.xyz formats). Survey stations may be referenced in scraps etc. When importing SurveX' 3D file, stations are inserted in survey hierarchy, if there exists identical hierarchy to that in 3D file.

Синтаксис:

```
import <file-name> [OPTIONS]
```

Контекст:

survey / all³⁸

Опции:

- `filter <prefix>` ▷ if specified, only stations with given prefix and shots between them will be imported. Prefix will be removed from station names.
- `surveys (create)/use/ignore` ▷ specifies how to import survey structure (works only with .3d files).
 - `create` ▷ split stations into subsurveys, if subsurveys do not exist, create them
 - `use` ▷ split stations into existing subsurveys
 - `ignore` ▷ do not split stations into sub-surveys
- `cs <coordinate system>` ▷ coordinate system for stations with fixed coordinates
- `calibrate [<x> <y> <z> <X> <Y> <Z>]` ▷ coordinates in the imported file are shifted from lower-case coordinates to upper-case coordinates.

'grade'

Описание:

This command is used to store predefined precisions of centreline data. See `sd` option description for `centreline` command.

Синтаксис: : `grade <id> ... [<quantity list> <value> <units>] ... endgrade`

Контекст:

all

³⁸ only with .3d files, where survey structure is specified

'revise'

Описание:

This command is used to set or change properties of an already existing object.

Синтаксис:

The syntax of this command for object created with "single line" command is

```
revise id [-option1 value1 -option2 value2 ...]
```

For objects created with "multi line" commands is syntax following

```
revise id [-option1 value1 -option2 value2 ...] ... optionX valueX data ... endrevise
```

Контекст:

all

Аргументы:

The id stands for object identifier (the id of an object you want to revise must always be specified).

Custom attributes

Objects *survey*, *centreline*, *scrap*, *point*, *line*, *area*, *map* and *surface* can contain user-defined attributes in a form `-attr <name> <value>`. `<name>` may contain alphanumeric characters, `<value>` is a string.

The custom attributes are used in map export depending on output format:

- in *shapefile* export they are written directly to the associated dbf file,
- in maps generated using METAPOST (PDF, SVG) the attributes are written in the METAPOST source file as strings (named like `ATTR_<name>`) and can be evaluated and used by user in symbols definition macros.

You can test presence of such a variable using `if known ATTR_<name>: ... fi`.

XTherion

XTherion – графический пользовательский интерфейс для Therion. Он помогает в создании файлов входных данных. В настоящее время он работает в трех основных режимах: текстовый редактор, редактор карт и компилятор.³⁹

³⁹ Здесь мы обсуждаем созданием данных, поэтому в этом разделе описаны только два первых режима. Функции компилятора смотрите в главе Обработка данных.

Его не обязательно использовать для Therion – 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. Можно написать лучшую, более удобную для пользователя, более WYSIWYG, 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. Right-click on the menu button and select in the menu which of the other menus it should be swapped with.

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.⁴⁰

Чтобы упростить ввод данных, он поддерживает форматирование таблички 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 отображаются как простые точки. Тип и другие атрибуты любого объекта указываются только в меню *Точка* и *Линия*.

⁴⁰ Кодировка файла указана в первой строке файла. Эта строка скрыта XTherion'ом и может быть доступна только косвенно, используя правое меню.

Упражнение: Найдите две существенные причины, почему карта, нарисованная в XTherion, не может быть идентична выходу Therion. (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.⁴¹



После создания нового файла в редакторе карт вы можете загрузить одно или несколько **изображений** – сканированные эскизы съемки пещеры⁴² – 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. Переместите 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)*.⁴³

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 ... end-scrap` может содержать команды `point`, `line` и `area`. (См. 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 ... end-scrap` многострочной команды. В меню *Команды в файле* выберите подменю

⁴¹ В одном файле можно нарисовать несколько скрапов, в этом случае все неактивные скрапы отображаются желтым.

⁴² 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.

⁴³ Примечание: Therion никак не использует фоновые изображения, если вы не назначили их для определенного скрапа с помощью опции `-sketch`.

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

```
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 мышью 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 мышью (отрегулируйте вторую контрольную

точку предыдущей дуги и первую контрольную точку следующего 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`⁴⁴, содержимое которого (введенное в меню *Редактор* в редакторе карты) обычно многострочная команда `area ... endarea` (см. раздел *Формат данных*.)

Если вы нарисуете несколько скрапов с `none` проекцией, необходимо **откалибровать** область рисования. Масштаб можно определить только одним способом в XTherion – 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.

⁴⁴ ВНИМАНИЕ! Команда `text` – это не команда Therion'a! Это всего лишь псевдоним для блока произвольного текста в XTherion. 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 '#'.

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.

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.⁴⁵ 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.

Дополнительные инструменты

5.3 **Помощь/Привязать изображение** создает файл MAP совместимый с OziExplorer на основе геоданных включенных в карту PDF⁴⁶.

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 преобразованием⁴⁷ в растровый формат.

⁴⁵ В случае станций съемки XTherion автоматически увеличивает номер станции для следующего вставленного символа.

⁴⁶ Может присутствовать до девяти калибровочных точек в виде фиксированных станций в нитке хода с использованием геодезической системы координат.

⁴⁷ Ghostscript и convert должны быть установлены в вашей системе. Обратите внимание, что установщик Windows не содержит ghostscript.

5.3 **Данные PocketToro** экспортированные в формате Therion'a⁴⁸ 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.

Сочитания клавишь и мыши в редакторе карт

Общие

- Ctrl+Z ▷ undo
- Ctrl+Y ▷ redo
- F9 ▷ compile current project
- to select object in the listbox using keyboard: switch using 'Tab' into desired listbox; move with underlined cursor to desired object; press 'Space'
- PageUp/PageDown ▷ scroll up/down in the side panel
- Shift+PageUp/PageDown ▷ scroll up/down in file commands window

Drawing area and background images

- RightClick ▷ scroll drawing area
- Double RightClick on the image ▷ move the image

Inserting scrap

- Ctrl+R ▷ insert scrap

Inserting line

- Ctrl+L ▷ insert new line and enter an 'insert line point' mode
- LeftClick ▷ insert line point (without control points)
- Ctrl+LeftClick ▷ insert line point very close to existing point (normally it's inserted right above closest existing point)
- LeftClick + drag ▷ insert line point (with control points)
- hold Ctrl while dragging ▷ fix the distance of previous control point
- LeftClick + drag on the control point ▷ move its position
- RightClick on one of the previous points ▷ selects the previous point while in insert mode (useful if you want to change also the direction of previous control point)

⁴⁸ Это специальный текстовый формат, который нужно импортировать с помощью XTherion'a и не может обрабатываться непосредственно Therion'ом.

- Esc or LeftClick on the last point ▷ end the line insertion
- LeftClick on the first line point ▷ close the line and end line insertion

Editing line

- LeftClick + drag ▷ move line point
- Ctrl+LeftClick + drag ▷ move line point close to the existing point (normally it is moved right above closest existing point)
- LeftClick on control point + drag ▷ move control point

Adding line point

- select the point before which you want to insert points; insert required points; press Esc or left-click on the point you selected at the beginning

Deleting line point

- select the point you want to delete; press *Edit line* → *Delete point* in the *Line control* panel

Splitting line

- select the point at which you want to split the line; press *Edit line* → *Split line* in the *Line control* panel

Inserting point

- Ctrl+P ▷ switch to 'insert point' mode
- LeftClick ▷ insert point at given position
- Ctrl+LeftClick ▷ insert point very close to existing point (normally it will be inserted right above the closest point)
- Esc ▷ escape from the 'inset point' mode

Editing point

- LeftClick + drag ▷ move point
- Ctrl+LeftClick + drag ▷ move point close to the existing point (normally it is moved right above closest existing point)
- LeftClick + drag on point arrows ▷ change point orientation or sizes (according to given switches in Point control panel)

Inserting area

- press Ctrl+A or *File commands* → *Insert* → *area* to switch to the 'insert area border' mode
- RightClick on the lines, that surround desired area
- Esc to finish area border lines insertion

Editing area

- select area you want to edit
- pres 'Insert' in the *Area control* to insert other border lines at current cursor position
- pres 'Insert ID' to insert border with given ID at current cursor position
- pres 'Delete' to remove selected area border line

Selecting an existing object

- LeftClick ▷ select object on the top
- RightClick ▷ select object right below the top object (useful when several points lie above each other)

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.⁴⁹ 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.

We usually insert maps in the last-but-one level in survey hierarchy.⁵⁰ Each scrap may then contain arbitrary part of any survey in the last level of hierarchy. 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](#)).⁵¹

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 именуется [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.

Можно создавать один файл [INDEX.th2](#) на каталог, который подключает все *.th2 файлы, описывает maps и объединения скрапов.

⁴⁹ Например [survey entrance](#) в файле [entry.th](#).

⁵⁰ Remember that surveys create namespaces, so you may reference only objects in the given survey and all subsurveys.

⁵¹ 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](#).

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`).

Therion in depth

How the map is put together

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

Экспортируются только видимые символы.

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

- *точечные символы*: символы наполнения ходов (почва... 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 файле⁵²: 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 внизу⁵³ заполняется `color preview-below`;
- Для каждого уровня⁵⁴:

НАЧАЛО прозрачности

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

⁵² Или в меню XTherion'a

⁵³ As specified using the `preview` option in the `map` command

⁵⁴ Уровень – это коллекция скрапов не разделенная `break` в команде `map`