



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.

We both step and do not step in the same rivers.
Ποταμοῖς τοῖς αὐτοῖς ἐμβαίνομέν τε καὶ οὐκ ἐμβαίνομεν.
—Heraclitus of Ephesus, 6th/5th century BC

Processing data

Besides data files, which contain survey data, Therion uses a configuration file, which contains instructions on how the data should be presented.

Configuration file

The configuration filename can be given as an argument to therion. By default Therion searches for file named `thconfig` in the current working directory. It is read like any other therion file (i.e. one command per line; empty lines or lines starting with '#' are ignored; lines ended with a backslash continue on the next line.) A list of currently supported commands follow.

'system'

Allows to execute system commands during therion compilation.² Normally Therion waits until the subprocess is finished. If you want to continue compilation without break, use `<command> &` syntax on Linux and `start <command>` syntax on Windows.

'encoding'

Works like the `encoding` command in data files—specifies character sets.

'language'

`n»„n»„n»„n»„n»„n»„n»„n»„:`

- `language <xx_[YY]>`

Sets the output language for translatable texts.

5.3

'cs'

`n»„n»„n»„n»„n»„n»„n»„n»„:`

- `cs <coordinate system>`

² E.g. to open or refresh external PDF viewer.

Outside of `layout` command specifies the coordinate system for output. It is not possible to specify more coordinate systems for different outputs (the last occurrence of `cs` is used for all output files). 5.3

If no `cs` is defined in the configuration file, the first `cs` therion encounters in the data files is used as an output `cs`.

Inside the `layout` specifies coordinate system for subsequent location data (`origin`, `grid-origin`).

'sketch-warp'

```
n»„n»„n»„n»„n»„n»„n»„n»„:
```

- `sketch-warp <algorithm>`

Specifies which scrap warping (morphing) algorithm to use. Possible algorithms are `line`—the default; `plaque`—invented by Marco Corvi.

'input'

Works like `input` command in data files—includes other files.

'source'

```
n»„n»„n»„n»„n»„n»„n»„n»„:
```

Specifies which source (data) files Therion should read. You can specify several files here; one per line. You can also specify them using the `-s` command line option (see below).

It is also possible to type (some small snippets of) code directly in configuration file using the multi-line syntax.

```
n»„n»„n»„n»„n»„n»„n»„n»„:
```

```
source <file-name>
```

```
or
```

```
source
```

```
...therion commands...
```

```
endsource
```

```
n»„n»„n»„n»„n»„n»„n»„n»„:
```

- `<file-name>`

'select'

`n»„n»„n»„n»„n»„n»„n»„n»„:`

selects objects (surveys and maps) for export. By default, all survey objects are selected. If there is no map selected, all scraps belonging to selected surveys are selected by default for map export.

If there are no scraps or maps in the data, centreline from all surveys is exported in the map.

When exporting maps in different projections, you need to select them for each projection separately.

`n»„n»„n»„n»„n»„n»„n»„n»„:`

`select <object> [OPTIONS]`

`n»„n»„n»„n»„n»„n»„n»„n»„:`

- `<object>` ▷ any survey or map, identified by its ID.

`n»„n»„n»„n»„n»„:`

- `recursive <on/off>` ▷ valid only when a survey is selected. If set on (by default) all subsurveys of the given survey are recursively selected/unselected.
- `map-level <number>` ▷ valid only when a map is selected. Determines the level at which map expansion for atlas export is stopped. By default 0 is used; if 'basic' is specified, expansion is done up to the basic maps. *Note:* Map previews are displayed only as specified in maps in the current `map-level`.
- `chapter-level <number>` ▷ valid only when a map is selected. Determines the level at which chapter expansion for atlas export is stopped. By default 0 is used, if '-' or '.' is specified, no chapter is exported for this map. If `title-pages` option in `layout` is on, each chapter starts with a title page.

'unselect'

`n»„n»„n»„n»„n»„n»„n»„n»„:`

Unselects objects from export.

`n»„n»„n»„n»„n»„n»„n»„n»„:`

`unselect <object> [OPTIONS]`

`n»„n»„n»„n»„n»„n»„n»„n»„:`

The same as the `select` command.

`n»„n»„n»„n»„n»„:`

The same as the `select` command.

'text'

n»„n»„n»„n»„n»„n»„n»„n»„:

Specifies translation of any default therion text in output.

n»„n»„n»„n»„n»„n»„n»„n»„n»„:

text <language ID> <therion text> <my text>

n»„n»„n»„n»„n»„n»„n»„n»„n»„:

- <language ID> ▷ standard ISO language identifier (e.g. `en` or `en_GB`)
- <therion text> ▷ therion text to translate. For list of therion texts and available translations, see `thlang/texts.txt` file.

'layout'

n»„n»„n»„n»„n»„n»„n»„n»„:

Specifies layout for 2D maps. Settings which apply to atlas mode are marked 'A'; map mode 'M'.

n»„n»„n»„n»„n»„n»„n»„n»„n»„:

layout <id> [OPTIONS] copy <source layout id> cs <coordinate system> north <true/grid>
scale <picture length> <real length> base-scale <picture length> <real length> units <met-
ric/imperial> rotate <number> symbol-set <symbol-set> symbol-assign <point/line/area/group/special>■
<symbol-type> \ <symbol-set> symbol-hide <point/line/area/group/special> <symbol-type>
symbol-show <point/line/area/group/special> <symbol-type> symbol-colour <point/line/area/group/special>■
<symbol-type> <colour> min-symbol-scale <scale> fonts-setup <tinysize> <smallsize> <normal-
size> <largesize> <hugesize> size <width> <height> <units> overlap <value> <units> page-
setup <dimensions> <units> page-numbers <on/off> exclude-pages <on/off> <list> title-pages
<on/off> nav-factor <factor> nav-size <x-size> <y-size> transparency <on/off> opacity <value>
surface <top/bottom/off> surface-opacity <value> sketches <on/off> layers <on/off> grid <off/top/bottom>■
grid-origin <x> <y> <x> <units> grid-size <x> <y> <z> <units> grid-coords <off/border/all>
origin <x> <y> <z> <units> origin-label <x-label> <y-label> own-pages <number> page-grid
<on/off> legend <on/off/all> legend-columns <number> legend-width <n> <units> map-comment■
<string> 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> <filename> statistics <explo/topo/carto/copyright■
all/off/number> <explo/topo-length on/off> scale-bar <length> <units> survey-level <N/all>
language <xx_YY> colour/color <item> <colour> debug <on/all/first/second/scrap-names/station-■
names/off> doc-author <string> doc-keywords <string> doc-subject <string> doc-title <string>
code <metapost/tex-map/tex-atlas> encode endlayout

n»„n»„n»„n»„n»„n»„n»„n»„n»„:

`<id>` ▷ layout identifier (to be used in the `export` command)

`n»„n»„n»„n»„n»„ n»„n»„n»„n»„n»„n»„n»„n»„ n»„n»„n»„n»„n»„:`

- `copy <source layout id>` ▷ set properties here that are not modified based on the given source layout.

map presentation-related:

- `scale <picture length> <real length>` ▷ set scale of output map or map atlas (M, A; default: 1 200)
- `base-scale <picture length> <real length>` ▷ if set, Therion will optically scale the map by a (`scale/base-scale`) factor. This has the same effect as if the map printed in `base-scale` would be photo-reduced to the `scale`. (M, A)
- `rotate <value>` ▷ rotates the map (M, A; default: 0)
- `units <metric/imperial>` ▷ set output units (M, A; default: `metric`)
- `symbol-set <symbol-set>` ▷ use `symbol-set` for all map symbols, if available. Be aware, that symbol set name is case sensitive. (M, A)

Therion uses following predefined symbol sets:

UIS (International Union of Speleology)

ASF (Australian Speleological Federation)

AUT (Austrian Speleological Association)

CCNP (Carlsbad Caverns National Park)

5.4

NZSS (New Zealand Symbol Set)

5.4

SKBB (Speleoklub Banska Bystrica)

- `symbol-assign <point/line/area/group/special> <symbol-type> <symbol-set>` ▷ display a particular symbol in the given symbol-set. This option overrides `symbol-set` option.

If the symbol has a subtype, `<symbol-type>` argument may have one of the following forms: `type:subtype` or simply `type`, which assigns new symbol set to all subtypes of a given symbol.

Following symbols may not be used with this option: point *section* (which isn't rendered at all) and all point and line labels (*label*, *remark*, *altitude*, *height*, *passage-height*, *station-name*, *date*). See the chapter *Changing layout/Customizing text labels* for details how to change labels' appearance. (M, A)

Group may be one of the following: all, centerline, sections, water, speleothems, passage-fills, ice, sediments, equipment.

5.3

5.4

There are two special symbols: north-arrow, scale-bar.

- `symbol-hide` `<point/line/area/group/special>` `<symbol-type>` ▷ don't display particular symbol or group of symbols.

You may use `group cave-centerline`, `group surface-centerline`, `point cave-station`, `point surface-station` and `group text` in `symbol-hide` and `symbol-show` commands.

5.4

Use `flag:<entrance/continuation/sink/spring/doline/dig>` as a `<symbol-type>` to hide stations with particular flags (e.g. `symbol-hide point flag:entrance`).

May be combined with `symbol-show`. (M, A)

- `symbol-show` `<point/line/area/group/special>` `<symbol-type>` ▷ display particular symbol or group of symbols. May be combined with `symbol-hide`. (M, A)

- 5.3 • `symbol-colo[u]r` `<point/line/area/group/special>` `<symbol-type>` `<colour>` ▷ change colour of particular symbol or group of symbols.³ (M, A)

- 5.4.1 • `min-symbol-scale` `<scale>` ▷ define minimal `<scale>`, from which points and lines are displayed on the map. E.g. for `min-symbol-scale M`, no points or lines scaled `S` and `XS` will be shown on the map. `<scale>` has the same format, as `scale` option for points and lines.

- 5.4.1 • `fonts-setup` `<tinysize>` `<smallsize>` `<normalsize>` `<largesize>` `<hugesize>` ▷ specify size of the text in points. `<normalsize>` applies to point label, `<smallsize>` applies to remark and all other point labels. Each of them may apply to line label according to its `-size` option.

The defaults are 8 10 12 16 24 for scales upto 1:100; 7 8 10 14 20 for scales upto 1:200; 6 7 8 10 14 for scales upto 1:500 and 5 6 7 8 10 for scales smaller than 1:500.

page layout related:

- `size` `<width>` `<height>` `<units>` ▷ set map size in the atlas mode. If not specified, it will be calculated from `page-setup` and `overlap`. In map mode applies iff `page-grid` is `on` (M, A; default: 18 22.2 cm)
- `overlap` `<value>` `<units>` ▷ set overlap size in paper units in the atlas mode or map margin in the map mode (M, A; default: 1 cm)
- `page-setup` `<dimensions>` `<units>` ▷ set page dimensions in this order: paper-width, paper-height, page-width, page-height, left-margin and top-margin. If not specified, it will be computed from `size` and `overlap` (A; default: 21 29.7 20 28.7 0.5 0.5 cm)
- `page-numbers` `<on/off>` ▷ turn on/off page numbering (A; default: `true`)
- `exclude-pages` `<on/off>` `<list>` ▷ exclude specified pages from cave atlas. The list may contain page numbers separated by a comma or dash (for intervals) e.g. `2,4-7,9,23` means, that pages 2, 4, 5, 6, 7, 9 and 23 should be omitted. Only the map

³ Note: colour change currently applies to pattern fills only if (1) output format is PDF and (2) METAPOST version is at least 1.000

pages should be counted. (Set `own-pages 0` and `title-pages off` to get the correct page numbers to be excluded.) Changes of `own-pages` or `title-pages` options don't affect page excluding. (A)

- `title-pages <on/off>` ▷ turn on/off title pages before each atlas chapter (A; default: `off`)
- `nav-factor <factor>` ▷ set atlas navigator zoom factor (A; default: `30`)
- `nav-size <x-size> <y-size>` ▷ set number of atlas pages in both directions of navigator (A; default: `2 2`)
- `transparency <on/off>` ▷ set transparency for the passages (underlying passages are also visible) (M, A; default: `on`)
- `opacity <value>` ▷ set opacity value (used if `transparency` is `on`). Value range is 0–100. (M, A; default: `70`)
- `surface-opacity <value>` ▷ set the surface bitmap opacity (used if `transparency` is `on`). Value range is 0–100. (M, A; default: `70`)
- `surface <top/bottom/off>` ▷ set the position of the surface bitmap above/below the map. (M, A; default: `off`)
- `sketches <on/off>` ▷ turn on/off displaying of morphed sketch bitmaps. (M, A; default: `off`)
- `layers <on/off>` ▷ enable/disable PDF 1.5 layers (M, A; default: `on`)
- `grid <off/bottom/top>` ▷ enable/disable grid (optionally coordinates' values may be also displayed) (M, A; default: `off`)
- `cs <coordinate system>` ▷ coordinate system for `origin` and `grid-origin`
- `north <true/grid>` ▷ specify default orientation of the map. By default, true (astronomical) north is used. It is ignored when used with local coordinate system.
- `grid-origin <x> <y> <x> <units>` ▷ set coordinates of grid origin (M, A)
- `grid-size <x> <y> <z> <units>` ▷ set grid size in real units (M, A; default is equal to scalebar size)
- `grid-coords <off/border/all>` ▷ specify where to label grid with coordinates. (M, A; default: `off`)
- `origin <x> <y> <z> <units>` ▷ set origin of atlas pages (M, A)
- `origin-label <x-label> <y-label>` ▷ set label for atlas page which has the lower left corner at the given origin coordinates. May be either a number or a character. (M, A; default: `0 0`)
- `own-pages <number>` ▷ set number of own pages added before the first page of automatically generated pages in atlas mode (currently required for correct page numbering) (A; default: `0`)

- `page-grid <on/off>` ▷ show pages key plan (M; default: `off`)

map legend related:

- `map-header <x> <y> <off/n/s/e/w/ne/nw/se/sw/center>` ▷ print map header at location specified by `<x>` `<y>`. Predefined map header contains some basic information about cave: name, scale, north arrow, list of surveyors etc. It is fully customizable (see the chapter *Changing layout* for details). `<x>` is easting (left-right on page). `<y>` is northing (up/down page). Ranges for `<x>` and `<y>` are -100–200. Lower-left corner of the map is `0 0`, upper-right corner is `100 100`. The header is aligned with the specified corner or side to this anchor point. (M; default: `0 100 nw`)
- `map-header-bg <on/off>` ▷ when on, background of map header is filled with background color (e.g. to hide map grid). (M; default: `off`)
- `map-image <x> <y> <n/s/e/w/ne/nw/se/sw/center> <filename>` ▷ include image specified by `<filename>` into map at location specified by `<x>` `<y>`. For coordinates and alignment details, see `map-header` specification.
- `legend-width <n> <units>` ▷ legend width (M, A; default: `14 cm`)
- `legend <on/off/all>` ▷ display list of used map symbols in the map header. If set to `all`, all symbols from the current symbol set are displayed. (M, A; default: `off`)
- `colo[u]r-legend <on/off>` ▷ turn on/off legend of map-fg colours when map-fg is set to altitude, scrap or map (M, A)
- `legend-columns <number>` ▷ adjusts the number of legend columns (M, A; default: `2`)
- `map-comment <string>` ▷ optional comment displayed at the map header (M)
- `statistics <explo/topo/carto/copyright all/off/number>` or
- `statistics <explo/topo-length on/hide/off>` ▷ display some basic statistics; if set to `off`, team members are sorted alphabetically; otherwise according to their contribution to exploration and surveying (M, A; default: `off`)
- `scale-bar <length> <units>` ▷ set the length of the scale-bar (M, A)
- `language <xx[_YY]>` ▷ set output language. Available languages are listed on the copyright page. See the *Appendix* if you want to add or customize translations. (M, A)
- `colo[u]r <item> <colour>` ▷ customize colour for special map items (map-fg, map-bg, preview-above, preview-below, label). Colour range is 0–100 for grayscale, [0–100 0–100 0–100] triplet for RGB colours.

For `map-fg`, you can use `altitude`, `scrap` or `map` as colours. In this case the map is coloured according to altitude, scraps or maps.

For `map-bg`, you can use `transparent` to omit page background completely.

For labels, you can switch colour `on/off`. If `on`, labels are coloured using the colour of associated scrap.

- `debug <on/all/first/second/scrap-names/station-names/off>` ▷ draw scrap in different stages of transformation in different colours to see how Therion distorts map data. See the description of `scrap` command for details. The points with distance changed most during transformation are displayed orange. If `scrap-names` is specified, scrap names are shown for each scrap, `station-names` displays name of each survey station.
- `survey-level <N/all>` ▷ `N` is the number of survey levels displayed next to the station name (M, A; default: 0).

PDF related:

- `doc-author <string>` ▷ set document author (M, A)
- `doc-keywords <string>` ▷ set document keywords (M, A)
- `doc-subject <string>` ▷ set document subject (M, A)
- `doc-title <string>` ▷ set document title (M, A)

customization:

- `code <metapost/tex-map/tex-atlas>` ▷ Add/redefine \TeX and \METAPOST macros here. This allows user to configure various things (like user defined symbols, map and atlas layout at one place &c.) See the chapter *Changing layout* for details.
- `endcode` ▷ should end the \TeX and \METAPOST sections

'setup3d'

`n»„n»„n»„n»„n»„n»„n»„n»„:`

- `setup3d <value>`

Temporary hack to set sampling distance in meters when generating piecewise linear 3d model from passage walls made of Bezier curves. 5.3

'sketch-colors'

`n»„n»„n»„n»„n»„n»„n»„n»„:`

- `sketch-colors <number-of-colors>`

This option can be used to reduce size of sketch bitmap images in maps. 5.4



'export'

$n \gg n \gg n \gg n \gg n \gg n \gg n \gg n \gg n$:

Exports selected surveys or maps.

$n \gg n \gg n \gg n \gg n \gg n \gg n \gg n \gg n$:

- `export <type> [OPTIONS]`

$n \gg n \gg n \gg n \gg n \gg n \gg n \gg n \gg n$:

- `<type>` ▷ The following export types are supported:

`model` ▷ 3D model of the cave
`map` ▷ one page 2D map
`atlas` ▷ 2D atlas in more pages
`cave-list` ▷ summary table of caves
`survey-list` ▷ summary table of surveys
`continuation-list` ▷ list of possible continuations
`database` ▷ SQL database with centreline

`n»„n»„n»„n»„n»„:`

common:

- `encoding/enc <encoding>` ▷ set output encoding
- `output/o <file>` ▷ set output file name. If no file name is given the prefix "`cave.`" is used with an extension corresponding to output format.

If the output filename is given and no output format is specified, the format is determined from the filename extension.

model:

- `format/fmt <format>` ▷ set model output format. Currently the following output formats are supported: `loch` (native format; default), `compass` (plt file), `survex` (3d file), `dxs`, `esri` (3d shapefiles), `vrml`, `3dmf` and `kml` (Google Earth).
- `enable <walls/[cave/surface]-centerline/splay-shots/surface/all>` and
- `disable <walls/[cave/surface]-centerline/splay-shots/surface/all>` ▷ selects which features to export, if the format supports it. Surface is currently exported in `therion` format only.
- `wall-source <maps/centerline/all>` ▷ set source data for passage wall modeling.

map/atlas:

- `format/fmt <format>` ▷ set map format. Currently `pdf`, `svg`, `xhtml`⁴, `survex`, `dxs`, `esri`⁵, `kml` (Google Earth), `xvi`⁶ and `bbox`⁷ for map; `pdf` for atlas are supported.
- `projection <id>` ▷ unique identifier that specifies the map projection type. (See the `scrap` command for details.)

If there is no map defined, all scraps in the given projection are exported.

⁴ SVG embedded in XHTML file which contains also legend

⁵ ESRI shapefiles. Multiple files are written to a directory with the specified filename.

⁶ Xtherion vector image. XVI images may be used in xtherion to draw in-scale maps. Scale (100 DPI image resolution is assumed) and grid-size from layout are used in export.

⁷ Text file containing geographic coordinates of lower-left and upper-right corners of the map area.

If there are no scraps with the specified projection then Therion will display centreline from selected surveys.

- `layout <id>` ▷ use predefined map or atlas layout.
- `layout-xxx` ▷ where `xxx` stands for other layout options. Using this you can change some layout properties directly within the export command.
- `encoding/enc <encoding>` ▷ set output encoding

common for lists:

- `format/fmt <format>` ▷ set continuation output format. Currently the following output formats are supported: `html` (default), `txt`, `kml`⁸ and `dbf`.

continuation-list:

- `attributes <(on)/off>` ▷ set whether to export user defined attributes in continuation list table.
- `filter <(on)/off>` ▷ set whether continuations without comment/text should be filtered out.

cave-list:

- `location <(on)/off>` ▷ set whether to export coordinates of cave entrances in the table.
- `surveys (on)/off` ▷ exports raw list of caves when set `off`. Otherwise survey structure with aggregated statistics is also displayed.

database:

- `format/fmt <format>` ▷ currently `sql` and `csv`
- `encoding/enc <encoding>` ▷ set output encoding

File formats summary:

| | <i>export type</i> | <i>available formats</i> |
|-----|--------------------|--|
| | model | loch, dxf, esri, compass, survex, vrml, 3dmf, kml |
| 5.3 | map | pdf, svg, xhtml, dxf, esri, survex, xvi, kml, bbox |
| | atlas | pdf |
| 5.4 | database | sql, csv |
| | lists | html, txt, kml, dbf |

Running Therion

Now, after mastering data and configuration files, we're ready to run Therion. Usually this is done from the command line in the data directory by typing

⁸ For cave-list and continuation-list.

The full syntax is

or

$n \gg n \gg n \gg n \gg n \gg n \gg n \gg n \gg n$:

$n \gg n \gg n \gg n \gg n \gg n$:

- 23

XTherion—compiler

XTherion makes it easier to run Therion especially on systems without a command line prompt. Compiler window is the default window of XTherion. To run Therion it's enough to open a configuration file and press 'F9' or 'Compile' button.

XTherion displays messages from Therion in the lower part of the screen. Each error message is highlighted and is hyperlinked to the source file where the error occurred.

After a first run there are activated additional menus *Survey structure* and *Map structure*. User may comfortably select a survey or map for export by double clicking on some of the items in the tree. Simple click in the *Survey structure* tree displays some basic information about the survey in the *Survey info* menu.

What we get?

Information files

Log file

Besides the messages from Therion and other programs used, the log file contains information about computed values of magnetic declination and meridian convergence, loop errors and scrap distortions.

Absolute loop error is $\sqrt{'x^2 + 'y^2 + 'z^2}$, where $'x$ is the difference between the identical start and end points of the loop before the error distribution measured along the x coordinate axis; similarly for y and z . Percentual loop error is calculated as *absolute error / loop length*. Average error is simple arithmetic average of all loop errors.

Scrap distortion is calculated using the distortion measure defined for all pairs of points (point symbols, points and control points of line symbols) in the scrap. The measure is calculated as $\frac{|d_a - d_b|}{d_b}$, where d_b is the distance of points before warping and d_a is the distance of points after warping. The maximal and average scrap distortions are calculated as a maximum or average of such measures applied to all pairs of points.

XTherion

Therion provides some basic facts about each survey (length, vertical range, N–S range, E–W range, number of shots and stations) if `-x` option is given. This information is displayed in XTherion, *Compiler* window, *Survey info* menu, when some survey from the *Survey structure* menu is selected.

SQL export

SQL export makes it easy to get very detailed and subtle information about centreline. It is a text file starting with tables declaration (where '?' stands in the following listing for a maximal value required by the column data)

```
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));
```

which is followed by a mass of SQL insert commands. This file may be loaded into any SQL database (after some database-dependent initialization, which may include running a SQL server and connecting to it, creating a database and connecting to it. A good idea is to start a transaction before loading this file, if database doesn't start a transaction automatically.) It's important to set-up database encoding to match the one specified in Therion [export database](#) command.

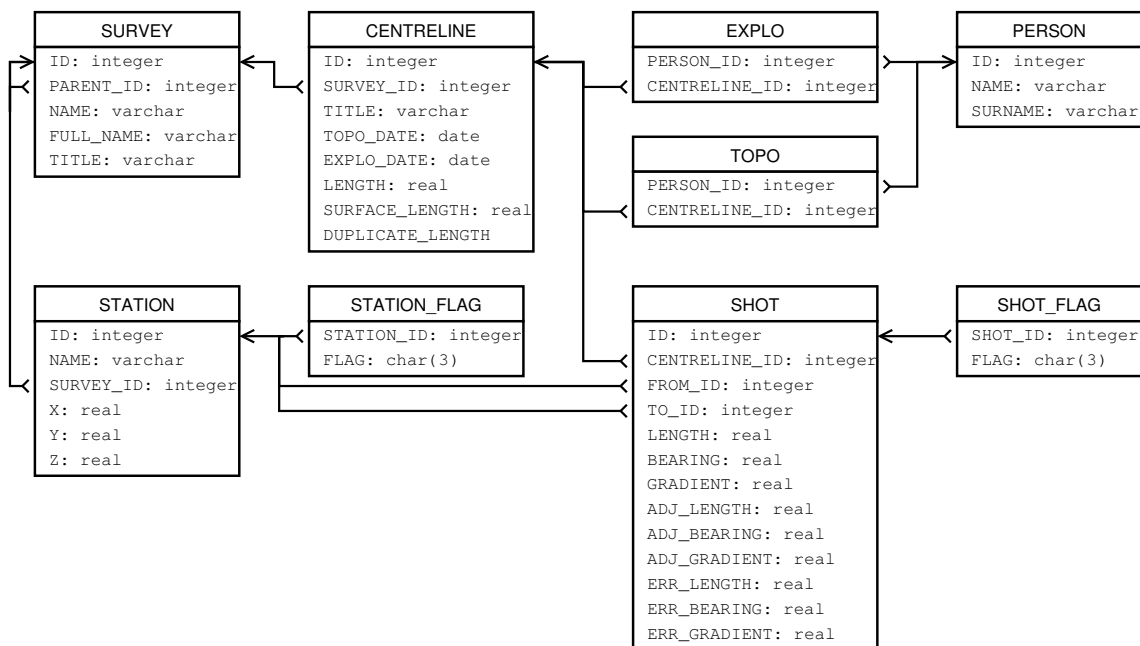


Table and column names are self-explaining; for undefined or non-existing values **NULL** is used. **FLAG** in **SHOT_FLAG** table is **dpl** or **srf** for duplicated or surface shots; in **STATION_FLAG** table **ent**, **con**, **fix**, **spr**, **sin**, **dol**, **dig**, **air**, **ove**, **arc** for stations with entrance, continuation, fixed, spring, sink, doline, dig, air-draught, overhang or arch attributes, respectively.

Examples of simple queries follow:

List of survey team members with an information how much has each of them surveyed:

```
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;
```

Which parts of the cave were surveyed in the year 1998?

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

How long are passages lying between 1500 and 1550 m a.s.l.?

```
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;
```

Lists—caves, surveys, continuations

Using `export continuation-list` you get an overview of all points in the centreline and scraps marked⁹ as a possible continuation.

`export cave-list` gives you a tabular information about surveyed caves (you need to specify `entrance` flags in your data) including length, depth and entrance(s) location.

Detailed information about each sub-survey gives `export survey-list` command. The length includes shots with `approximate` flags, but not `explored`, `duplicate` or `surface`.

2D maps

Maps for printing

Maps are produced in PDF and SVG formats, which may be viewed or printed in a wide variety of viewers. Be sure to uncheck *Fit page to paper* or similar option if you want to print in the exact scale.

In atlas mode some additional information is put on each page: page number, map name, and page label.

Especially useful are the numbers of neighbouring pages in N, S, E and W directions, as well as in upper and lower levels. There are also hyperlinks at the border of the map if the cave continues on the next page and on the appropriate cells of the Navigator.

PDF files are highly optimized—scraps are stored in XObject forms only once in the document and then referenced on appropriate pages. Therion uses most advanced PDF features like transparency and layers.

⁹ using `station` attribute for centreline point and `point continuation` in scraps

Created PDF files may be optionally post-processed in applications like pdfTeX or Adobe Acrobat—it's possible to extract or change some pages, add comments or encryption, etc.

- 5.3 If the map was produced using georeferenced data then it also contains georeferencing information. This can be extracted by XTherion to produce georeferenced raster images (see *XTherion/Additional tools* for details).

Maps for GIS

Maps produced in DXF, ESRI or KML formats may be further processed in appropriate software. These maps do not contain visualized map symbols

Special-purpose maps

Map in XVI format contains centreline with LRUD (and optionally morphed sketches) and can be imported in XTherion to serve as a background for digitization.

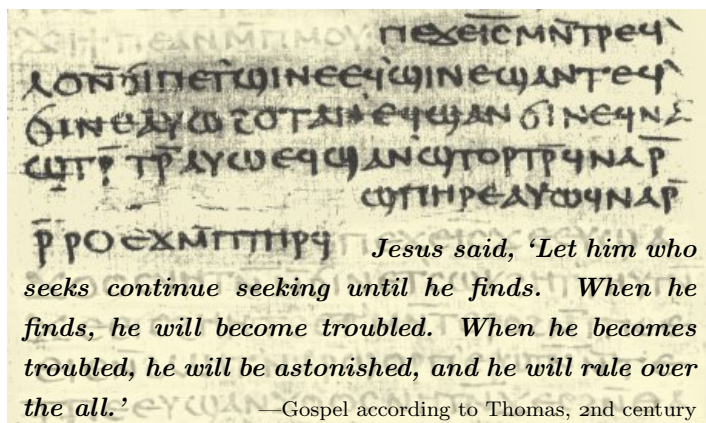
Map in Survox format is intended for a quick preview in Aven.

3D models

Therion may export 3D model in various formats besides its native format. These may be loaded in appropriate viewing, editing or raytracing programs to be printed or further processed. If the format doesn't support arbitrary passage shape definition, only the centreline is included.

Loch

Loch is a 3D model viewer included in the Therion distribution. It supports e.g. high-resolution rendering to file and stereo view using 3D-glasses.



Changing layout of PDF maps

This chapter is extremely useful if you're not satisfied with the predefined layout of map symbols and maps provided, and want to adapt them to your needs. However, you need to know how to write plain T_EX and METAPOST macros to do this.

Page layout in the atlas mode

The `layout` command allows basic page setup in the atlas mode. This is done through its options such as `page-setup` or `overlap`. But there are no options which would specify the position of map, navigator and other elements inside the area defined by `page-width` and `page-height` dimensions; e.g., why is the navigator below the map and not on its right or left side?

There are many possible arrangements for a page. Rather than offer even more options for the `layout` command, Therion uses the T_EX language to describe other page layouts.

This approach has the advantage that the user has direct access to the advanced typesetting engine without making the language of Therion overcomplex.

Therion uses pdfT_EX with the *plain* format for typesetting. So you should be familiar with the plain T_EX if you wish to define new layouts.

The ultimate reference for plain T_EX is

Knuth, D. E.: *The T_EXbook*, Reading, Massachusetts, Addison-Wesley ¹1984

For pdfT_EX's extensions there is a short manual

Thanh, H. T.—Rahtz, S.—Hagen, H.: *The pdfT_EX user manual*, available at <http://www.pdfTeX.org>

The \TeX macros are used inside of `code tex-atlas` part of the `layout` command (see the chapter *Processing data* for details). The basic one predefined by Therion is the

`\dopage`

macro. The idea is simple: for each page Therion defines \TeX variables (count, token, and box registers) which contain the page elements (map, navigator, page name etc.). At the end of each page macro `\dopage` is invoked. This defines the position of each element on the page. By redefining this macro you'll get desired page layout. Without this redefinition you'll get the standard layout.

Here is the list of variables defined for each page:

Boxes:

- `\mapbox` ▷ The box containing the map. Its width (height) is set according to the `size` and `overlap` options of the `layout` command to

`size_width + 2*overlap` or

`size_height + 2*overlap`, respectively

- `\navbox` ▷ The box containing the navigator, with dimensions

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

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

Both `\mapbox` and `\navbox` also contain hyperlinks.

Count registers:

- `\pointerE`, `\pointerW`, `\pointerN`, `\pointerS` contain the page number of the neighbouring pages in the E, W, N and S directions. If there is no such a page its page number is set to 0.
- `\pagenum` current page number

Token registers:

- `\pointerU`, `\pointerD` contain information about pages above and below the current page. It consists of one or more concatenated records. Each record has a special format

`page-name|page-number|destination|`

If there are no such pages, the value is set to `notdef`.

See the description of the `\processpointeritem` macro below for how to extract and use this information.

- `\pagename` ▷ name of the current map according to options of the `map` command.
- `\pagelabel` ▷ the page label as specified by `origin` and `origin-label` options of the `layout` command.

The following variables are set at the beginning of the document:

- `\hsize`, `\vsize` ▷ \TeX page dimensions, set according to `page-width` and `page-height` parameters of the `page-setup` option of the `layout` command. They determine our playground when defining page layout using the `\dopage` macro.
- `\ifpagenumbering` ▷ This conditional is set true or false according to the `page-numbers` option of the `layout` command.

There are also some predefined macros which help with the processing of `\pointer*` variables:

- `\showpointer` with one of the `\pointerE`, `\pointerW`, `\pointerN` or `\pointerS` as an argument displays the value of the argument. If the value is 0 it doesn't display anything. This is useful because the zero value (no neighbouring page) shouldn't be displayed.
- `\showpointerlist` with one of the `\pointerU` or `\pointerD` as an argument presents the content of this argument. (Which contains `\pointerU` or `\pointerD`, see above.) For each record it calls the macro `\processpointeritem`, which is responsible for data formatting.

Macro `\showpointerlist` should be used without redefinition in the place where you want to display the content of its argument; for custom data formatting redefine `\processpointeritem` macro.

- `\processpointeritem` has three arguments (page-name, page-number, destination) and visualizes these data. The arguments are delimited as follows

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

An example definition may be

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

(note how to use the *destination* argument), or much simpler (if we don't need hyperlink features):

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

For font management there are macros

- `\size[#1]` for size changes,
- `\color[#1 #2 #3]` for colour changes (RGB values in the range 0–100), and
- `\rm`, `\it`, `\bf`, `?`, `\si` for type face switching.

See below for a list of predefined texts which may be used in the atlas.

There is also a `\framed` macro which takes a box as an argument and displays the box framed. The frame style can be customized by redefining the `\linestyle` macro which defaults to `1 J 1 j 1.5 w`.

Now we're ready to define the `\dopage` macro. You may choose which of the predefined elements to use. A very simple example would be

```
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
```

which defines the landscape A4 layout without the navigator nor any texts. There is only a map on the page.

Note the `\insertmaps` macro. Map pages are inserted at its position. This is not done automatically because you may wish to insert some other pages before the first map page.

More advanced is the default definition of the `\dopage` macro:

```
\def\dopage{ \vbox{\centerline{\framed{\mapbox}} \bigskip

%
\line{ \vbox to \ht\navbox{ \hbox{\size[20]\the\pagelabel \ifpagenumbering\space(\the\pagenum)\fi
\space\size[16]\the\pagename} \ifpagenumbering

%
\medskip

\hbox{\qqquad\qqquad \vtop{ \hbox to 0pt{\hss\showpointer\pointerN\hss} \hbox to 0pt{\llap{\showpointer\poi
\raise1pt\hbox to 0pt{\hss$\updownarrow$\hss} \raise1pt\hbox to 0pt{\hss$\leftrightharpoonup$\hss}
\rlap{\hskip0.7em\showpointer\pointerE}} \hbox to 0pt{\hss\showpointer\pointerS\hss} } \qqquad\qqquad
\vtop{ \def\arr{$\uparrow$} \showpointerlist\pointerU \def\arr{$\downarrow$} \showpointerlist\pointerD
} } \fi

%%%%
\vss

\scalebar

}\hss \box\navbox } } }
```

Using other plain \TeX macros or \TeX primitives it's possible to add other features, e.g. a different layout for odd and even pages; headers and footers; or adding a logo to each page.

In addition to map pages contains atlas additional items: title page, basic facts about the cave, legend with used map symbols etc.

Therion automatically generates list of used map symbols and lists of persons who have discovered, surveyed and drawn selected part of the cave. Following token registers may be used (according to user's requirements before or after the `\insertmaps` macro):

- `\explotitle`, `\topotitle`, `\cartotitle` ▷ translated titles

- `\exploteam`, `\topoteam`, `\cartoteam` ▷ participating members (according to `team`, `explo-team` options for `centreline` and `author` option of `scraps`)
- `\explodate`, `\topodate`, `\cartodate` ▷ corresponding dates
- `\comment` ▷ is set according to `map-comment` option of the `layout` command
- `\copyrights` ▷ is set according to copyright options for surveys and other objects
- `\cavename` ▷ name of the exported map; set according to `-title` option of exported map
- `\cavelength`, `\cavedepth` ▷ approximate length and depth of displayed map
- `\cavelengthtitle`, `\cavedepthtitle` ▷ translated labels
- `\cavemaxz`, `\caveminz` ▷ altitude max/min value 5.4
- `\thversion` ▷ current therion version 5.4
- `\currentdate` ▷ current date 5.4
- `\outscode`, `\outsname` ▷ output coordinat system code and name 5.4
- `\northdir` ▷ 'true' or 'grid' 5.4
- `\magdecl` ▷ magnetic declination in degrees 5.4
- `\gridconv` ▷ grid meridian convergence in degrees 5.4

There is a macro `\atlastitlepages` which combines most of the token registers mentioned above to get simple preformatted atlas introductory pages.

For legend displaying there are

- `\iflegend` ▷ conditional; true iff `legend` option of the `layout` command was set to `on` or `all` values
- `\legendtitle` ▷ token register containing translated legend title
- `\insertlegend` ▷ macro for inserting legend symbols pictures with translated descriptions in the specified number of columns (according to `legend-columns` layout option)
- `\formattedlegend` ▷ combines all three above commands to get preformatted legend with header and symbols typeset in two¹⁰ columns if `legend` option is set `on`

North arrow and scale bar may be displayed using

- `\ifnortharrow` ▷ conditional; true if map projection is plan and symbol north-arrow is not hidden in `layout`
- `\ifscalebar` ▷ conditional; true if scalebar is not hidden
- `\northarrow` ▷ PDF form with the north arrow
- `\scalebar` ▷ PDF form with the scale bar

¹⁰ Default; adjust the `legend-columns` layout option to get them more or less

There is a general-purpose macro for typesetting in multiple columns¹¹:

- `\begmulti <i>`, `\endmulti >` text between these macros is typeset in `<i>` columns

Example how to create atlas with lists of surveyors etc. followed by map pages and with legend at the end:

```
code tex-atlas \atlastitlepages
```

```
\insertmaps
```

```
\formattedlegend
```

Page layout in the map mode

In the map mode it's possible to use a lot of predefined variables which are described in the previous chapter:

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

In order to place them somewhere on the map page, you have to define `\maplayout` macro in the `code tex-map` section of the `layout` command. It should contain one or more `\legendbox` invocations. The `\legendbox` macro has four parameters: coordinates ranging 0–100, alignment specification (N, E, S, W, NE, SE, SW, NW or C) and the content to be displayed.

A simple example is

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

which displays north arrow in the upper-left corner of the map sheet.

For user's convenience, there is `\legendcontent` token register. It contains preformatted cave name, north arrow, scale bar, explo/topo/carto teams, comment, copyrights and legend. (The `\legendcontent` is also used in the default map layout definition: `\def\maplayout{\legendbox{0}{100}{NW}{\the\legendcontent}}`).

Width of the above text may be adjusted by `\legendwidth` dimen register (its default value is set by `legend-width` layout option). The color and size of texts in the preformatted legend can be easily changed using `\legendtextcolor`, `\legendtextsize`, `\legendtextsectionsize` and `\legendtextheadersize` token registers, e.g. for large blue text:

¹¹Not to be used with map legend, where multiple columns are to be adjusted by `legend-columns` layout option

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

It is possible to display the whole map framed by setting the `\framethickness` dimension register to positive value, e.g. `0.5mm`.

Customizing text labels

Starting with the release 5.4.1 you can use `fonts-setup` layout option instead of the METAPOST macro `fonts_setup()`.

New map symbols

Therion's layout command makes it easy to switch among various predefined map symbol sets. If there is no such symbol or symbol set you want, it's possible to design new map symbols.

However, this requires knowledge of the METAPOST language, which is used for map visualization. It's described in

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

User may also benefit from comprehensive reference to the METAFONT language, which is quite similar to METAPOST:

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

New symbols may be defined in the `code metapost` section of the `layout` command. This makes it easy to add new symbols at the run-time. It is also possible to add symbols permanently by compiling them into Therion executable (see the *Appendix* for instructions how to do this).

Each symbol has to have a unique name, which consists of following items:

- one of the letters 'p', 'l', 'a', 's' for point, line, area or special symbols, respectively;
- underscore character;
- symbol type as listed in the chapter *Data format* with all dashes removed;
- if the symbol has a subtype, add underscore character and subtype;
- underscore character;
- symbol set identifier in uppercase

Example: standard name for a point 'water-flow' symbol with a 'permanent' sub-type in the 'MY' set is `p_waterflow_permanent_MY`. Standard name for user-defined symbol types should not include symbol set identifier, e.g. `p_u_bat`.

Each new symbol has to be registered by a macro call

```
initsymbol("<standard-name>");
```

unless it's compiled into Therion executable.

There are four predefined pens *PenA* (thickest) ... *PenD* (thinnest), which should be used for all drawings. For drawing and filling use `thdraw` and `thfill` commands instead of METAPOST's `draw` and `fill`.

5.4 The following variables are also available:

- boolean `ATTR__shotflag_splay`, `ATTR__shotflag_duplicate`,
`ATTR__shotflag_approx` ▷ set for line survey
- boolean `ATTR__stationflag_splay` ▷ set true for endstations of splay shots
- boolean `ATTR__scrap_centerline` ▷ set true for scraps created from centreline
- boolean `ATTR__elevation` ▷ true for (extended) elevation, false for plan projection
- numeric `ATTR__height` ▷ height of a pit or wall:pit
- string `ATTR__id` ▷ contains current object ID
- string `ATTR__survey` ▷ contains current survey name
- string `ATTR__scrap` ▷ contains current scrap name
- picture `ATTR__text` ▷ contains typeset text e.g. for point continuation
- string `NorthDir` ▷ 'true' or 'grid'
- numeric `MagDecl` ▷ magnetic declination in degrees
- numeric `GridConv` ▷ grid meridian convergence in degrees

Point symbols

Point symbols are defined as macros using `def ... enddef` commands. Majority of point symbol definitions has four arguments: position (pair), rotation (numeric), scale (numeric) and alignment (pair). Exceptions are *section* which has no visual representation; all *labels*, which require special treatment as described in the previous chapter, and *station* which takes only one argument: position (pair).

All point symbols are drawn in local coordinates with the length unit u . Recommended ranges are $\langle -0.5u, 0.5u \rangle$ in both axes. The symbol should be centered at the coordinates' origin. For the final map, all drawings are transformed as specified in the T transformation variable, so it's necessary to set this variable before drawing.

This is usually done in two steps (assume that four arguments are P, R, S, A):

- set the U pair variable to $\left(\frac{width}{2}, \frac{height}{2}\right)$ of the symbol for correct alignment. The alignment argument A is a pair representing ratios $\left(\frac{shift_x}{U_x}\right)$ and $\left(\frac{shift_y}{U_y}\right)$.

(Hence `aligned A` means `shifted (xpart A * xpart U, ypart A * ypart U)`.)

- set the T transformation variable

`T:=identity aligned A rotated R scaled S shifted P;`

For drawing and filling use `thdraw` and `thfill` commands instead of METAPOST's `draw` and `fill`. These take automatically care of T transformation.

An example definition may be

```
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");
```

Line symbols

Line symbols differ from point symbols in respect that there is no local coordinate system. Each line symbol gets the *path* in absolute coordinates as the first argument. Therefore it's necessary to set T variable to `identity` before drawing.

Following symbols take additional arguments:

- `arrow` ▷ numeric: 0 is no arrows, 1 arrow at the end, 2 begin, 3 both ends
- `contour` ▷ text: list of points which get the tick or one of -1 , -2 or -3 to mark undefined tick, tick in the middle or no tick, respectively
- `section` ▷ text: list of points which get the orientation arrow or -1 to indicate no arrows
- `slope` ▷ numeric: 0 no border, 1 border; text: list of (point,direction,length) triplets

Usage example:

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

Area symbols

Areas are similar to lines: they take only one argument – *path* in absolute coordinates.

You may fill them in three ways:

- fill an uniform or randomised grid in a temporary picture (having dimensions `bbox path`) with some point symbols; clip it according to `path` and add to the `currentpicture`

- fill `path` with a solid colour
- fill `path` with a predefined pattern using a `withpattern` keyword.

Patterns are defined using the same user interface (without the `patterncolor` macro) as described in the article

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

You may use standard METAPOST `draw` and similar macros without setting of T variable in pattern definitions.

Example on how to define and use patterns:

```
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");
```

Special symbols

There are currently two special symbols: scale bar and north arrow. Both are experimental and subject to change.

1. *When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong.*
2. *The only way of discovering the limits of the possible is to venture a little way past them into the impossible.*
3. *Any sufficiently advanced technology is indistinguishable from magic.*

—Arthur C. Clarke, 1973

Appendix

Compilation

If you want to compile Therion from source code and run it, you need (first three are required only during compilation):

- GNU C/C++ compiler
- GNU make
- Perl
- Python 2.7 or 3
- Tcl/Tk 8.4.3 and newer (<https://www.tcl.tk>) with *BWidget* widget set (<https://sourceforge.net/projects/tcllib/>) and optionally *tkImg* extension (<https://sourceforge.net/projects/tkimg/>).
- T_EX distribution with at least T_EX with Plain format, recent pdfT_EX, and META-POST (<https://www.tug.org>).
- LCDF Typetools package (<https://www.lcdf.org/type/>)
- ImageMagick distribution with *convert* and *identify* utilities, if you want to use warping of survey sketches.
- *ghostscript* if you want to create calibrated images from georeferenced PDF maps.

To compile Loch, you need

- freetype 2 and newer; freetype-config must work
- wxWidgets 2.6 and newer; wx-config must work
- VTK 5.0 and newer
- libjpeg, libpng, zlib

All programs (with the exception of BWidget and tkImg package) are usually included in Linux, Unix or MacOS X distributions. For Windows consider using MinGW and MSYS (<http://www.mingw.org>). It's a distribution of GNU utilities with GNU make and GCC. (BTW, why not to use precompiled Windows version?)

Quick start

- unpack the source distribution [therion-5.*.tar.gz](#)
- `cd therion`
- `make config-macosx` or `make config-win32`, if you use MacOS X or Windows, respectively
- `make`
- `sudo make install`

Hacker's guide

Make parameters

Therion's *makefile* may take some optional parameters.

- `config-linux`, `config-macosx`, `config-win32` ▷ configure Therion for a specific platform. Linux is a default.
- `config-release`, `config-oxygen`, `config-ozone` ▷ set optimization level for C++ compiler (none, `-O2` and `-O3`)
- `config-debug` ▷ useful before debugging the program
- `install` ▷ install Therion
- `clean` ▷ delete all temporary files

5.4 *Cross-compilation for Windows*

Therion supports compilation of Win32 executables in Linux using MXE cross compiler (<http://mxe.cc>).

- install the following static/win32 packages (i686-w64-mingw32.static-*) to the directory `/usr/lib/mxe/`: binutils, bzip2, expat, freetype-bootstrap, gcc, gettext, glib, harfbuzz, jpeg, libiconv, libpng, tiff, vtk, wxwidgets, xz, zlib.
- modify PATH: `export PATH=/usr/lib/mxe/usr/bin:$PATH`
- `cd therion`
- `make config-win32cross`
- `make`

Adding new translations

Therion supports translation of map labels. Suppose you want to add a new language `xx`.

- run `'perl process.pl export xx'` in the `'thlang'` Therion source subdirectory. This creates a file `texts_xx.txt`. This file is UTF-8 encoded.
- edit the `texts_xx.txt` file. Add your translations at lines beginning with `'xx:'`.
- run `make update`
- compile Therion

Adding new encodings

Although UTF-8 Unicode encoding covers all characters which Therion is able to process, it may be inconvenient to use it. In that case it's possible to add support for any 8-bit encoding for text input files. Copy a translation file to the `thchencdata` directory; add its name to `'ifiles'` hash in the beginning of the Perl script `generate.pl`; run it and recompile Therion.

The translation file should contain two hexadecimal values of a character (first one in the 8-bit encoding, second one in Unicode) in each line. Possible comments follow the `'#'` character.

Adding new T_EX encodings

It's easy to add new encodings for 2D map output.¹² Copy an appropriate encoding mapping file with an `*.enc` extension to the `texenc/encodings`, run the Perl script `mktexenc.pl` located in the `texenc` directory and compile Therion.

Therion uses the same encoding files as `afm2tfm` program from the T_EX distribution, which has the same format as an encoding vector in a PostScript font. You may find more details in the chapter 6.3.1.5 *Encoding file format* in the documentation to Dvips program.

Generating new T_EX and METAPOST headers

Therion uses T_EX and METAPOST for 2D map visualization and typesetting. Pre-defined macros are compiled into the Therion executable and are copied to the working directory just before running METAPOST and T_EX (unless the `-use-externlibs` option is used). Layout command makes it possible to modify some macros in the configuration file at the run-time.

However, it's possible to make permanent changes to the macro files. After modifying the files in the `mpost` and `tex` directories it's necessary to run Perl scripts `genmpost.pl` and `gentex.pl`, which generate C++ header files, and compile Therion executable again.

¹² This section applies to old-style font selection using `tex-fonts` command in the initialization file and is obsolete when using `pdf-fonts` command. 5.3

Environment variables

Therion reads following environment variables:

- **THERION** ▷ [not required] search path for (x)therion.ini file(s)
- **HOME** (**HOMEDRIVE** + **HOMEPATH** on WinXP) ▷ [not required, but usually present on your system] search path for (x)therion.ini file(s)
- **TEMP**, **TMP** ▷ system temporary directory, where Therion stores temporary files (in a directory named **th\$PID\$**, where **\$PID\$** is a process ID), unless **tmp-path** is specified in the initialization file.

Consult the documentation of your OS how to set them.

Initialization files

Therion's and XTherion's system dependent settings are specified in the file **therion.ini** or **xtherion.ini**, respectively. They are searched for in the following directories:

- on UNIX: **.**, **\$THERION**, **\$HOME/.therion**, **/etc**, **/usr/etc**, **/usr/local/etc**
- on Windows: **.**, **\$THERION**, **\$HOME\therion**, **<Therion-installation-directory>**, **C:\WINDOWS**, **C:\WINNT**, **C:\Program Files\Therion**

Therion

If no file is found Therion uses its default settings. If you want to list them, use **-print-init-file** option. The initialization file is read like any other therion file. (Empty lines or lines starting with **#** are ignored; lines ending with a backslash continue on next line.) Currently supported initialization commands follow.

- **loop-closure <therion/survex>**

By default, **survex** is used if present, otherwise **therion**.

- **encoding-default <encoding-name>**

Set the default output encoding (currently unused).

- **encoding-sql <encoding-name>**

Set the default output encoding for SQL export.

- **language <xx[_YY]>**

Default output language. See the copyright page for the list of available languages.

- `units <metric/imperial>`
Set default units.
- `mpost-path <file-path>`
Set the full path to a METAPOST executable if Therion can't find it ("`mpost`" is the default).
- `mpost-options <string>`
Set METAPOST options.
- `pdftex-path <file-path>`
Set the full path to a pdfT_EX executable if Therion can't find it ("`pdfetex`" is the default).
- `identify-path <file-path>`
Set the full path to ImageMagick's identify executable if Therion can't find it ("`identify`" is the default).
- `convert-path <file-path>`
Set the full path to ImageMagick's convert executable if Therion can't find it ("`convert`" is the default).
- `source-path <directory>`
Path to data and configuration files. Used mostly for system-wide grades and layout definitions.
- `tmp-path <directory>`
Path where temporary directory should be created.
- `tmp-remove <OS command>`
System command to delete files from the temporary directory.
- `tex-env <on/off>`
[Works on Windows only.] When set to `off` (default), Therion temporarily clears all environment variables related to T_EX. Useful if there is other T_EX distribution installed on your system which had set-up any environment variables, which could confuse T_EX and METAPOST programs supplied in Therion for Windows distribution.

Set to `on` if you use other T_EX distribution for maps processing.
- `text <language ID> <therion text> <my text>`
Using this option you can change any default therion text translation in output. For list of therion texts and available translations, see `thlang/texts.txt` file.
- `cs-def <id> <proj4def>`

Define a new coordinate system <id> using Proj4 syntax.

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

5.3

Set-up fonts to be used in PDF maps. The command has to be followed by paths specifying where regular, italic, bold, sans-serif and sans-serif oblique fonts are located in your system. TrueType and OpenType fonts are supported.

Therion requires LCDF Typetools to be installed on your system to use this command. Example:

```
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"
```

- 5.3
- `otf2pfb <on/off>`

When set to `on` (default), OpenType fonts used in `pdf-fonts` are converted to PFB fonts, if they are PostScript-based. Some information is lost in the PFB format, but there is advantage that pdfTeX can embed subset of PFB fonts (in contrast with OpenType fonts which must be fully embedded).

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

Original and more complicated way to set-up fonts for PDF maps. You need to explicitly specify encoding (maximum 256 characters from the font that will be used). The list of currently supported encodings gives the `-print-tex-encodings` command line option. The same encoding must be used while generating TeX metrics (*.tfm files) for those fonts (e.g. with the `afm2tfm` program) and this encoding must be explicitly given also in the pdfTeX's map file. The only exception is the base set of Computer Modern fonts, which use 'raw' encoding. This encoding doesn't need to be specified in the pdfTeX's map file.

Encoding has to be followed by five font specifications for regular, italic, bold, sans-serif and sans-serif oblique styles. Default setting is `tex-fonts raw cmr10 cmti10 cmbx10 cmss10 cmssi10`

Example how to use other fonts (e.g. TrueType Palatino in `xl2` (an encoding derived from ISO8859-2) encoding). Run:

```
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
```

You get files `vpalatino.vf`, `vpalatino.tfm` and `palatino.tfm`. Add the line

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

to the pdfTeX's map file. The same should be done for the italic and bold faces and corresponding sans-serif and sans-serif-oblique fonts. If you're lazy try

`tex-fonts xl2 palatino palatino palatino palatino palatino`

(We should use actually virtual font `vpalatino` instead of `palatino`, which contains no kerning or ligatures, but pdfT_EX doesn't support `\pdfincludechars` command on virtual fonts. To be improved.)

If you want to add some unsupported encodings, read the chapter *Compilation / Hacker's guide*.

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

Similar to `tex-fonts`, but tests if the T_EX fonts are installed in the system. It does nothing if any of the specified fonts is not present.

This setting is used by default for Czech/Slovak and cyrillic fonts to avoid METAPOST errors on systems without these fonts present.

As the test takes some time (pdfT_EX instance is run), you might disable the default behaviour completely by setting `tex-fonts` in the INI file.

XTherion

Initialization file for XTherion is actually a Tcl script evaluated when XTherion starts. The file is commented; see the comments for details.

Limitations

- scrap size $\triangleright \approx 2.8 \times 2.8$ m in the output scale (METAPOST limit)
- page size \triangleright
PDF map or atlas: $\approx 5 \times 5$ m (pdfT_EX limit)
SVG map: unlimited
- scraps count \triangleright approx. 500–6000, depending on frequency of cross-sections
current METAPOST limit: $4(scrap + sections) < 4096$ (may be arbitrarily increased)
pdfT_EX limit: $2 \times pages + images + patterns + 6(scrap + sections) < 32500$

Example data

Following simple example illustrates basic usage of Therion commands:

```
encoding utf-8
```

```
survey main -title "Test cave"
```

```
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
```

```
# scraps are usually in separate *.th2 files scrap s1 -author 2004 "Therion team"
```

```
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 endlane
```

```
endscrap
```

```
map m1 -title "Test map" s1 endmap
```

```
endsurvey
```

Corresponding configuration file could be:

```
encoding utf-8 source test
```

```
layout l1 scale 1 100 layers off endlayout
```

```
select m1@main
```

```
export model -fmt survex export map -layout l1
```

If you save data file as 'test.th' and configuration file as 'thconfig' you may process them with Therion.

History

- 1999

Oct: first concrete ideas

Nov: start of programming (Perl scripts and METAPOST macros)

Dec 27: Therion compiles simple map in PostScript format for the first time (32 kB of Perl and 7 kB of METAPOST and T_EX source code). The map warping model was substantially different from the current one (positions of features were relative to a particular survey shot, not to positions of all stations in a scrap). This version already included some interesting features such as *transformation functions* which allowed user specification of the input format for survey data, or splitting large maps to multiple sheets.

Dec 30: the first web page (with data examples but without source code)

- **2000**

Jan: xthedit (Tcl/Tk), a graphical front-end for Therion

Feb 18: start of reprogramming (Perl)

Apr 1: the first hyperlinked PDF cave map / atlas

Aug: experiments with PDF, pdfT_EX and METAPOST

- **2001**

Nov: start of reimplementation from scratch: Therion (C++ with some Perl scripts inherited from the previous version); notion of a scrap; interactive 2D map editor ThEdit as a replacement of xthedit (Delphi)

Dec: ThEdit exports simple map for the first time

- **2002**

Mar: Therion 0.1 — Therion is able to process survey data (centreline) of the Cave of Dead Bats. XTherion, text editor designed for Therion (Tcl/Tk).

Jul 27: Therion 0.2 — Therion compiles simple map (consisting of two scraps) for the first time (800 kB of source code)

Aug: XTherion extended to 2D map editor (as a replacement of ThEdit)

Sep: Therion compiles first real and complex map of a cave. XTherion extended to compiler.

- **2003**

Mar: the first version of The Therion Book finished

Apr: Therion included in Debian GNU/Linux

Jun: all Perl scripts rewritten in C++, Therion is one executable program now (although using Survex and T_EX)

- **2004**

Mar: Therion 0.3 — Therion exports 3D model created from 2D maps. Loop closure algorithm included into Therion.

- **2006**

Oct: Therion 0.4 — New 3D viewer (Loch).

- **2007**

Feb: Therion 0.5 — Support for bitmap sketches morphing.

Future

Although Therion is already used for map production, there are a lot of new features to be implemented:

General

- loop closure information in SQL

2D maps

- complete the predefined symbol sets
- generate registers for atlas
- use MPlib instead of METAPOST

3D models

- improve passage walls modeling

XTherion

- improve 2D editing capabilities

Loch

- colour schemes
- survey tree for selecting sub-surveys to display
- spatial filtering (e.g. clipping by planes)
- support for multiple surfaces

Labyrinth

- completely new GUI in the far future (see <https://labyrinth.speleo.sk>)