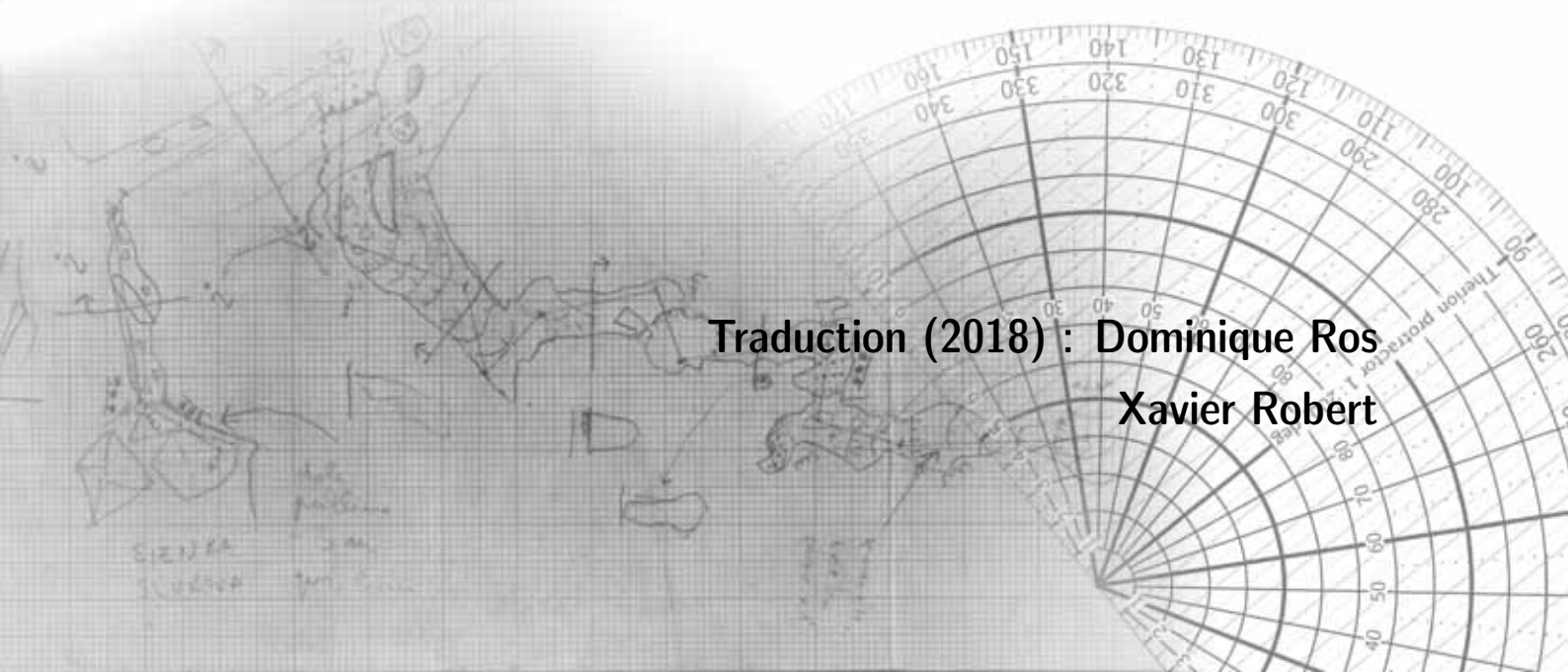


La Bible Therion

Stacho Mudrák

Martin Budaj



Traduction (2018) : Dominique Ros

Xavier Robert

Therion est un logiciel protégé par le droit d’auteur, distribué sous la licence publique générale GNU.

Copyright © 1999–2018 Stacho Mudrák, Martin Budaj

Ce livre décrit Therion 5.4.1+? (compiled on 2018-06-19).

Contributions au code de *Olly Betts*, *Marco Corvi*, *Vladimir Georgiev*, *Georg Pacher* et *Dimitrios Zachariadis*.

Nous remercions :

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

et tous les utilisateurs pour leurs retours/commentaires, leur soutien et leurs suggestions.

Traductions (%):

<i>Langue</i>	<i>XTherion</i>	<i>Map header</i>	<i>Loch</i>	<i>Traduit par</i>
bg	86	87	100	Alexander Yanev, Ivo Tachev, Vladimir Georgiev
ca	–	100	–	Evaristo Quiroga
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, Evaristo Quiroga
fr	?	87	?	Eric Madelaine, Gilbert Fernandes, Xavier Robert
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

L’illustration de couverture montre un croquis de la salle de *Hrozný kameňolom* dans la “Grotte des chauves-souris mortes” (Cave of Dead Bats) en Slovaquie et la topographie de celle-ci produite par Therion.

Table des matières

Introduction	5
Pourquoi Therion?	5
Caractéristiques	6
Logiciels requis	7
Installation	8
Environnement de travail	8
Comment cela fonctionne-t-il ?	8
Première utilisation	9
Création de fichiers de données	10
Les bases	10
Types de données	12
Systèmes de coordonnées	13
Déclinaison magnétique automatique ou manuelle	14
Format des données	14
‘encoding’	15
‘input’	15
‘survey’	145
‘centreline’	17
‘scrap’	22
‘point’	24
‘line’	28
‘area’	31
‘join’	32
‘equate’	33
‘map’	33
‘surface’	34
‘import’	36
‘grade’	37
‘revise’	37
Attributs personnalisés	38
XTherion	38
XTherion—module éditeur de texte	39
XTherion—module éditeur de dessin	39
Outils additionnels	43
Raccourcis clavier et souris dans l’éditeur de dessins	44
Penser Therion	46
Comment entrer une ligne de cheminement ?	46
Comment dessiner des topographies ?	47
Comment créer des modèles ?	48
Therion en profondeur	48
Comment le dessin est-il construit ?	48
Traitement des données	51
Fichier de configuration	44
‘system’	44
‘encoding’	44
‘language’	44
‘cs’	44
‘sketch-warp’	45
‘input’	45

‘source’	45
‘select’	46
‘unselect’	46
‘text’	47
‘layout’	47
‘setup3d’	53
‘sketch-colors’	53
‘export’	53
Faire fonctionner Therion	56
XTherion—compiler	57
Qu’obtenons-nous ?	59
Fichiers d’information	59
Fichier Log	59
XTherion	59
Export SQLt	59
Listes—cavités, topographies, continuations	61
Plans 2D	61
Plans à imprimer	61
Plans pour SIG	62
Plans pour applications spécifiques	62
Modèles 3D	62
Loch	62
Mise en page des dessins en PDF	63
Mise en page en mode Atlas	63
Mise en page en mode Topographie (cartographie)	68
Personnalisation des étiquettes de texte	69
Nouveaux symboles topographiques	70
Symboles de points	71
Symboles de lignes	72
Symboles de surfaces	72
Symboles spéciaux	73
Annexes	74
Compilation	74
Démarrage rapide	75
Guide du Hacker	75
Variables d’environnement	77
Fichiers d’initialisation	77
Therion	77
XTherion	80
Limitations	80
Exemple de données	80
Historique	82
Futur	83
Général	83
Dessins 2D	84
Modèles 3D	84
XTherion	84
Loch	84
Labyrinth	84

Introduction

Therion est un outil de topographie souterraine. Son but est d'aider à :

- archiver les données topographiées sur ordinateur sous une forme aussi proche que possible des notes et des esquisses d'origine, et de les récupérer de manière flexible et efficace ;
- dessiner et actualiser de beaux plans et coupes ;
- créer un modèle 3D réaliste de la grotte.

Il fonctionne avec les systèmes d'exploitation Unix, Linux, MacOS X et Win32. Le code source et le programme d'installation de Windows sont disponibles sur la page Internet de Therion (<https://therion.speleo.sk>).

Therion est distribué sous licence [GNU General Public License](#).

Pourquoi Therion ?

Dans les années 90, nous avons beaucoup pratiqué la spéléologie et la topographie des grottes. Il existait certains programmes informatiques affichant des vues de cheminement et des stations après la fermeture de la boucle et l'élimination des erreurs. Ceux-ci ont été d'une grande aide, en particulier pour les systèmes de cavités de grande taille et complexes. Nous avons utilisé l'un d'eux, TJKPR, comme couche de fond avec le calque des stations pour dessiner ensuite des topographies à la main. Après avoir achevé un vaste atlas de la Grotte des chauves-souris de 166 pages au début de 1997, nous avons rapidement eu un problème : nous avons trouvé de nouveaux passages reliant des passages connus et les avons explorés. Après traitement dans TJKPR, les nouvelles boucles ont influencé la position des anciens relevés. La plupart des stations du relevé avaient maintenant une position légèrement différente devant en raison de la modification de la répartition des erreurs. Ainsi, nous devions soit dessiner à nouveau tout l'Atlas, soit accepter que l'emplacement de certains endroits ne soit pas précis (dans le cas de boucles d'une longueur d'environ 1 km, il y avait parfois des erreurs d'environ 10 m) et d'essayer de fausser les nouveaux passages pour les adapter aux anciens.

Ces problèmes ont persisté lorsque nous avons essayé de dessiner des topographies à l'aide de certains programmes de CAO en 1998 et 1999. Il était toujours difficile d'ajouter de nouveaux relevés sans adapter les anciens aux positions nouvellement calculées des stations topo dans toute la grotte. Nous n'avons trouvé aucun programme capable de dessiner une topo complexe correctement mise à jour (cest-à-dire pas seulement des vues

avec une enveloppe LRUD), dans laquelle les anciennes parties soient modifiées en fonction des coordonnées connues des stations du relevé le plus récent.

En 1999, nous avons commencé à réfléchir à la création d'un programme propre pour le dessin de topographies. Nous connaissions des programmes parfaitement adaptés à des sous-tâches particulières. Il y avait **METAPOST**, un langage de programmation de haut niveau pour la description de graphiques vectoriels, *Survex* pour un excellent traitement des plans, et *TeX* pour la compilation des résultats. Nous n'avions qu'à les rassembler. À Noël 1999, une version minimaliste de Therion fonctionnait pour la première fois. Cela ne comprenait que 32 kB de scripts Perl et de macros **METAPOST**, mais visait à démontrer que nos idées pouvaient être mises en œuvre.

En 2000-2001, nous avons recherché le format optimal des données d'entrée, le langage de programmation, le concept d'éditeur de dessins interactif et des algorithmes internes avec l'aide de Martin Sluka (Prague) et de Martin Heller (Suisse). En 2002, nous avons pu sortir la première version de Therion réellement utilisable, qui répondait à nos exigences.

Caractéristiques

Therion est une application en ligne de commande. Il traite les fichiers d'entrée, y compris les topos 2D, au format texte, et crée des fichiers avec des topos 2D ou un modèle 3D en sortie.

La syntaxe des fichiers d'entrée est décrite en détail dans les chapitres suivants. Vous pouvez créer ces fichiers dans un éditeur de texte brut quelconque tel que *ed* ou *vi*. Ils contiennent des instructions pour Therion comme :

```
point 1303 1004 pillar
```

où **point** est un mot-clé pour le symbole de point, suivi de ses coordonnées et d'une spécification de type de symbole à afficher (ici un pilier).

L'édit manuel de tels fichiers n'est pas facile - en particulier lorsque vous tracez des topos, devez penser en termes spatiaux (coordonnées cartésiennes). Il existe donc une interface graphique spéciale pour Therion appelée XTherion. XTherion fonctionne comme un éditeur de texte avancé, un éditeur de topos (où les images sont dessinées de manière totalement interactive) et un compilateur (qui exécute Therion sur les données).

Cela peut paraître compliqué, mais cette approche présente de nombreux avantages :

- Il existe une séparation stricte des données et de la visualisation. Les fichiers de données spécifient uniquement ce qui est où, pas ce à quoi cela ressemble. **METAPOST** ajoute la représentation visuelle dans les phases ultérieures du traitement des données. (Ce concept est très similaire à la représentation de données XML.)

Cela permet de modifier les symboles de dessin utilisés sans modifier les données d'entrée ou de fusionner plusieurs dessins créés par différentes personnes ayant des styles différents dans un seul dessin avec un jeu de symboles unifié.

Les dessins en 2D sont adaptés à des échelles de sorties spécifiques (niveau d'abstraction, mise à l'échelle non linéaire des symboles et des textes)

- Toutes les données sont relatives aux positions des stations de la topographie. les coordonnées des stations de la topographie sont modifiées au cours du processus de fermeture de la boucle, toutes les données pertinentes sont déplacées en conséquence, de sorte que la topographie est toujours à jour.
- Therion est indépendant des systèmes d'exploitation, d'un encodage de caractères ou d'un éditeur de fichiers d'entrée particulier ; les fichiers d'entrée resteront lisibles par un être humain.
- Il est possible d'ajouter de nouveaux formats de sortie.
- Le modèle 3D est généré à partir des images 2D pour obtenir un modèle 3D réaliste sans entrer trop de données.
- bien que le support pour WYSIWYG soit limité, vous obtenez ce que vous voulez.

Logiciels requis

“Un programme doit faire une seule chose, mais le faire bien.” (Ken Thompson) Nous utilisons donc des programmes externes utiles, liés aux problèmes de composition et de visualisation des données. Therion peut alors s'acquitter de sa tâche beaucoup mieux que si il s'agissait d'une application autonome permettant de calibrer votre imprimante ou votre scanner et d'envoyer un courrier électronique avec vos données d'un seul clic.

Therion a besoin de :

- Une distribution $\text{T}_{\text{E}}\text{X}$. Nécessaire uniquement si vous souhaitez créer des dessins 2D au format PDF ou SVG.
- *Tcl* / *Tk* avec *BWidget* et éventuellement l'extension *tkImg*. Ce n'est nécessaire que pour *XTherion*.
- Des outils typographiques *LCDF* (*LCDF Typetools*) si vous souhaitez utiliser une configuration facile pour les polices personnalisées dans les dessins PDF.
- *Convertissez* et *identifiez* les utilitaires à partir de la distribution *ImageMagick*, si vous souhaitez utiliser la déformation des esquisses de topographie.
- *ghostscript* pour créer des images calibrées à partir de fichiers PDF géoréférencés.

Le programme d'installation Windows inclut tous les packages requis, à l'exception de *ghostscript*. Lisez l'*Annexe* si vous voulez compiler vous-même Therion.

Pour afficher des dessins et des modèles, vous pouvez utiliser l'un des programmes suivants :

- tout visualisateur PDF ou SVG affichant des images 2D ;
- tout SIG prenant en charge les formats DXF ou *shapefile* pour l'analyse des dessins ;

- une visionneuse 3D appropriée pour les modèles exportés dans un format autre que le format par défaut;
- tout client de base de données SQL pour traiter la base de données exportée.

Installation

Installation à partir des sources (package `therion-5.*.tar.gz`):

Le code source est une distribution primaire de Therion. Il doit être compilé et installé conformément aux instructions de l'*annexe*.

Installation pour Windows:

Exécutez le programme d'installation et suivez les instructions.

Il installe tous les éléments requis et crée des raccourcis vers XTherion et Therion Book.

Environnement de travail

Therion lit les paramètres du fichier d'initialisation. Les paramètres par défaut devraient fonctionner correctement pour les utilisateurs utilisant uniquement les caractères latins¹, T_EX standard et METAPOST.

Si vous souhaitez utiliser vos propres polices pour les caractères latins ou non latins dans les fichiers PDF, modifiez le fichier d'initialisation. Des instructions sur la manière de procéder sont données en *annexe*.

Comment cela fonctionne-t-il ?

Alors, maintenant que les besoins de Therion sont clairs, voyons comment il interagit avec tous ces programmes :

PAS DE PANIQUE ! Lorsque votre système est correctement configuré, la plus grande partie est cachée à l'utilisateur et tous les programmes nécessaires sont automatiquement exécutés par Therion de façon transparente.

Pour travailler avec Therion, il suffit de savoir que vous devez créer des données d'entrée (la meilleure méthode étant avec XTherion), puis exécuter Therion et afficher les fichiers de sortie (modèle 3D, dessins, fichier journal) dans le programme approprié.

Pour en savoir plus à ce sujet, voici une brève explication de l'organigramme ci-dessus. Les noms de programme sont en police romaine et les fichiers de données en italique.

¹ Sur le fichier PDF, Therion restitue la plupart des caractères accentués sous la forme d'une combinaison d'accent et d'un caractère de base. Certains accents étranges peuvent être omis. Des lettres accentuées précomposées sont incluses pour les langues slovaque et tchèque.



Les flèches indiquent le flux de données entre les programmes. Les fichiers de données temporaires ne sont pas affichés. Signification des couleurs :

- noir—Programmes Therion et macros (XTherion est écrit Tcl/Tk, il a donc besoin de cet interpréteur pour fonctionner)
- rouge—package T_EX
- vert—fichiers entrants créés par l'utilisateur et fichiers sortants créés par Therion

Therion effectue lui-même la tâche principale. Il lit les fichiers d'entrée, les interprète, trouve les boucles fermées et distribue les erreurs. Ensuite, toutes les autres données (par exemple, dessins 2D) sont transformées en fonction de la nouvelle position de la station. Therion exporte les données des dessins 2D au format METAPOST. METAPOST donne la forme réelle aux symboles de carte abstraits en fonction des définitions de symbole de carte ; cela crée beaucoup de fichiers PostScript avec de petits scraps de la grotte. Ceux-ci sont lus et convertis dans un format de type PDF, qui forme les données d'entrée pour pdfT_EX. pdfT_EX effectue ensuite toute la composition et crée le fichier PDF final de la topographie de la grotte.

Therion exporte aussi le modèle 3D (complet ou ligne de cheminement) dans différents formats.

Le cheminement peut être exporté pour un traitement ultérieur dans n'importe quelle base de données SQL.

Première utilisation

Après avoir expliqué les principes de base de Therion, il est judicieux de lessayer sur des exemples de données.

- Téléchargez les exemples de données à partir de la page Web Therion et décompressez-les quelque part sur le disque dur de votre ordinateur.

- Exécutez XTherion (sous Unix et MacOSX en tapant ‘xTherion’ dans la ligne de commande, vous trouverez sous Windows un raccourci dans le menu *Démarrer*).
- Ouvrez le fichier ‘thconfig’ à partir du répertoire de données de la fenêtre ‘Compiler’ de XTherion.
- Appuyez sur ‘F9’ ou ‘compile’ dans le menu pour exécuter Therion sur les données— Vous obtiendrez des messages de Therion, METAPOSTet T_EX
- Les fichiers PDF et le modèle 3D sont créés dans le répertoire de données.

De plus, vous pouvez ouvrir des fichiers de données topo (*.th) dans la fenêtre ‘Editeur de texte’ et des fichiers de données cartographiques (*.th2) dans la fenêtre ‘Editeur de carte’ de XTherion. Bien que le format des données puisse sembler déroutant la première fois, il vous sera expliqué dans les chapitres suivants.

Nous n'avons écrit ce travail que pour vous, enfants de doctrine et d'apprentissage. Examinez ce livre, réfléchissez au sens que nous avons dispersé et rassemblé à nouveau ; ce que nous avons caché à un endroit, nous l'avons révélé à un autre, afin que cela puisse être compris par votre sagesse.

Vos igitur doctrinæ & sapientiæ filii, perquirite in hoc libro, colligendo nostram disperfam intentionē, quam in diuersis locis proposuimus, & quod occultatum est à nobis in uno loco, manifestum fecimus illud in alio, ut sapientibus uobis patefiat, uobis enim solis scripsimus

—Henricus C. agrippa ab Nettesheym, 1533

Création de fichiers de données

Les Bases

Les fichiers d'entrée pour Therion sont au format texte. Il y a quelques règles de base sur la forme d'un tel fichier :

- Il existe deux types de commandes : Commandes sur une ligne et commandes sur plusieurs lignes.
- Une commande sur une seule ligne est terminée par un caractère de fin de ligne. La syntaxe de celle-ci est :

```
command arg1 ... argN [-option1 value1 -option2 value2 ...]
```

où *arg1 ... argN* sont des arguments obligatoires et les paires commençant par un tiret *-option value* sont des options facultatives. Les arguments et les options disponibles dépendent de la commande. Un exemple pourrait être :

```
point 643.5 505.0 gradient -orientation 144.7
```

avec trois arguments obligatoires (au début) et une paire optionnelle 'option / valeur' (après le tiret). Les options peuvent n'avoir aucune ou plusieurs valeurs.

- Les commandes multi-lignes commencent de la même manière que les commandes à une seule ligne, mais continuent sur les lignes suivantes jusqu'à la fin de la commande explicite (endcommande). Ces lignes peuvent contenir des données ou des options, qui sont appliquées aux données suivantes. Si une ligne de données commence par un mot réservé à une option, vous devez insérer '!' Devant celle-ci. La syntaxe est :

```
command arg1 ... argN [-option1 value1 -option2 value2 ...]
...
optionX valueX
data
...
endcommand
```

Encore une fois, pour une meilleure compréhension, voici un exemple réel avec la commande "line" (ligne) :

```
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
```

Cette commande line a un argument obligatoire, un type de ligne (paroi de cavité dans ce cas), suivi d’une option. Les deux lignes suivantes contiennent des données (coordonnées des courbes de Bézier à tracer). La ligne suivante (“smooth off” pour lisser) spécifie une option qui s’applique aux données suivantes (c’est-à-dire pas pour la ligne entière, contrairement à l’option `-id` dans la première ligne) et la dernière ligne contient quelques données supplémentaires.

- si la valeur d’une option ou d’un argument contient des espaces, vous devez inclure cette valeur entre " " ou []. Si vous souhaitez insérer un double guillemet " texte inclus " ", vous devez l’insérer deux fois. Les guillemets sont utilisés pour les chaînes, les crochets pour les valeurs numériques et les mots-clés (keywords).
- chaque ligne se terminant par une barre oblique inversée ou anti-slash (\) est considérée comme continuant sur la ligne suivante, comme s’il n’y avait ni saut de ligne ni barre oblique inversée.
- tout ce qui suit # (dièse), jusqu’à la fin de la ligne, même à l’intérieur d’une commande, est considéré comme un commentaire et est donc ignoré.

- 5.4 • les commentaires multilignes dans un bloc `comment ... endcomment` sont autorisés dans les fichiers de données et de configuration

Types de données

Therion utilise les types de données suivants :

- *keyword/mot-clef* ▷ une séquence de caractères `a-Z`, `a-z`, `0-9` et `_` (ne commençant pas par ‘-’).
 - *ext_keyword/mot-clef_étendu* ▷ mot-clé pouvant également contenir des caractères `+*,.,’,` Mais pas sur la première position.
 - *date* ▷ une date (ou un d’intervalle de temps) dans un format spécifique : `YYYY.MM.DD@HH:MM:SS.SS - YYYY.MM.DD@HH:MM:SS.SS` ou ‘-’ pour laisser une date indéterminée.
 - *person/personne* ▷ nom et prénom d’une personne séparés par des espaces. Utilisez ‘/’ pour séparer le prénom et le nom s’il y a plusieurs noms.
- 5.3 • *string/chaîne de caractère* ▷ une séquence de caractères quelconques. Les chaînes peuvent contenir une balise spéciale `<lang:XX>` pour séparer les traductions. Dans les chaînes multilingues, seul le texte compris entre `<lang:XX>` (où `XX` est la langue

sélectionnée dans le fichier d'initialisation ou de configuration) et la balise `<lang:YY>` suivante est affiché à la sortie. Si aucune correspondance n'est trouvée, tout ce qui précède toute balise `<lang:ZZ>` est affiché.

- *units* ▷ unités de longueur prises en charge : meter[s] (mètres), centimeter[s] (centimètres), inch[es] (pouces), feet[s] (pieds), yard[s] (aussi m, cm, in, ft, yd). Unités d'angle prises en charge : degree[s], minute[s] (aussi deg, min), grad[s], mil[s], percent[age] (uniquement pour le clinomètre). Une valeur en degrés peut être entrée en notation décimale (*x.y*) ou en notation spéciale pour les degrés, minutes et secondes (*deg[:min[:sec]]*).

Systèmes de coordonnées

Therion prend en charge les transformations de coordonnées dans les systèmes de coordonnées géodésiques. Vous pouvez spécifier l'option `cs` (CS pour **C**oordinate **S**ystem) dans les objets `centreline`, `surface`, `import` and `layout` (ligne de cheminement, surface, importation et mise en page), puis saisir des données XY dans un système donné. Vous pouvez également spécifier la sortie `cs` dans le fichier de configuration.

Si vous ne spécifiez pas de `cs` dans votre jeu de données, il est supposé que vous travaillez dans un système de coordonnées local et aucune conversion n'est effectuée. Si vous spécifiez `cs` n'importe où dans les données, vous devez le spécifier pour toutes les données de géolocalisation (`fix`, `origin` in `layout` etc.).

`cs` s'applique à toutes les données d'emplacement suivantes jusqu'à ce qu'un autre `cs` soit spécifié ou sinon jusqu'à la fin de l'objet en cours, selon celui qui est indiqué en premier.

Les systèmes de coordonnées suivants sont pris en charge :

- `UTM1` – `UTM60` ▷ Universal Transverse Mercator pour l'hémisphère nord et dans une zone donnée, WGS84 datum.
- `UTM1N` – `UTM60N` ▷ idem à `UTM1` – `UTM60`
- `UTM1S` – `UTM60S` ▷ UTM pour l'hémisphère sud, WGS84 datum.
- `lat-long`, `long-lat` ▷ latitude (N positive, S négative) et longitude (E positive, W négative) dans un ordre donné en degrés (*deg[:min[:sec]]* autorisé), WGS84 datum. Par défaut, non pris en charge en sortie.
- `EPSG:<nombre>` ▷ La plupart des systèmes de coordonnées EPSG. Presque tous les systèmes de coordonnées utilisés dans le monde ont leur propre numéro EPSG. Pour trouver le numéro de votre système, voir le fichier `extern/proj4/nad/epsg` dans la distribution de la source Therion.
- `ESRI:<nombre>` ▷ Similaire à EPSG, mais standard ESRI.
- `JTSK`, `iJTSK` ▷ Système tchécoslovaque S-JTSK utilisé depuis les années 1920 avec les axes sud et ouest (JTSK) et sa version modifiée avec les axes orientés est et nord et

les nombres négatifs (iJTSK). JTSK n'est pas pris en charge en sortie (mais iJTSK lui l'est).

- **JTSK03**, **iJTSK03** ▷ nouvelle réalisation S-JTSK introduite en Slovaquie en 2011.
- **OSGB:<H, N, O, S or T><a-Z except I>** ▷ Grille nationale du British Ordnance Survey.
- **S-MERC** ▷ la projection sphérique de Mercator, utilisée par divers sites de cartographie en ligne.

5.4

Déclinaison magnétique automatique ou manuelle

5.4

Therion intègre le modèle de champ géomagnétique Terre IGRF², valable pour la période 1900-2020. Il est utilisé automatiquement pour le calcul de la déclinaison magnétique si la cavité est située dans l'espace avec une station fix (géolocalisée) utilisant l'un des systèmes de coordonnées géodésiques pris en charge en même temps que la commande date des lignes de cheminement. La déclinaison calculée est alors répertoriée dans le fichier journal pour plus d'informations (LOG file).

Si l'utilisateur définit une **declination** (déclinaison magnétique) spécifique pour le cheminement, cette valeur a priorité sur le calcul automatique.

Format des données

La syntaxe des fichiers d'entrée est expliquée dans la description des commandes individuelles. L'étude des exemples de fichiers distribués avec Therion vous aidera à comprendre. Voir aussi un exemple en *annexe*.

Chacune des sections suivantes décrit une commande Therion à l'aide de la structure suivante : The syntax of input files is explained in the description of

Description : notes concernant cette commande.

Syntaxe : description de la syntaxe schématique.

Contexte : : spécifie le contexte dans lequel cette commande est autorisée. Le contexte de relevé topo (*survey*) signifie que la commande doit être entourée par la paire **survey ... endsurvey** pair. Le contexte de *scrap* signifie que la commande doit être incluse dans la paire **scrap ... endscrap**. Context *all* signifie que la commande peut être utilisée n'importe où.

Arguments : une liste des arguments obligatoires avec leurs explications.

Options : une liste des options utilisables.

Command-like options : options pour les commandes multi-lignes, qui peuvent être spécifiées parmi les lignes de données.

² Voir <https://www.ngdc.noaa.gov/IaGa/vmod/>

‘encoding’

Description : Définit l’encodage du fichier d’entrée. Cela permet d’utiliser d’autres caractères que les caractères ASCII dans les fichiers d’entrée.

Syntaxe : `encoding <nom-encodage>`

Contexte : Ce devrait être la toute première commande du fichier.

Arguments :

- `<encoding-name>` ▷ pour voir une liste de tous les noms de codage pris en charge, exécutez Therion avec l’option `--print-encodings`. ‘UTF-8’ (Unicode) et ‘aSCII’ (7 bit) sont toujours supportés.

‘input’

Description : Insère le contenu d’un fichier à la place de la commande. L’extension par défaut est ‘.th’ et peut être omise. Pour une compatibilité optimale, utilisez les chemins relatifs et les barres Unix ‘/’, et non les barres obliques inverses Windows ‘\’, comme séparateurs de répertoires.

Syntaxe : `input <file-name>`

Contexte : all

Arguments :

- `<nom-du-fichier>`

‘survey’

Description : Survey est la structure de données principale. Les topographies peuvent être imbriquées, ce qui permet de construire une structure hiérarchique. Habituellement, chaque niveau de cette étude de structure hiérarchique représente des cavités, des zones karstiques de niveaux supérieurs et de niveaux inférieurs, par exemple des galeries.

Chaque topographie a son propre espace de noms spécifié par son argument `<id>`. Les objets (tels que les stations topos ou les scraps ; voir ci-dessous qui appartiennent à une sous-topographie (subsurvey) de la topographie en cours sont référencés comme suit :

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

ou, s’il y a plusieurs niveaux de classement :

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

Cela signifie que les identificateurs d'objet ne doivent être uniques que dans le cadre d'une seule et même topographie. En conséquence, les noms des stations topographiques peuvent être identiques si elles se trouvent dans des topographies différentes. Cela permet aux stations d'être numérotées à partir de 0 dans chaque relevé ou d'assembler deux cavités dans un même système karstique sans renommer les stations du relevé.

Syntaxe : `survey <id> [OPTIONS]
... other therion objects ...
endsurvey [<id>]`

Contexte : aucun, survey

Arguments :

- `<id>` ▸ identifiant de la survey/topographie

Options :

- `namespace <on/off>` ▸ spécifie si survey crée un espace pour les noms (`on` par défaut)
- `declination <specification>` ▸ définit la déclinaison magnétique qui sera utilisée par défaut pour toutes les données des objets de cette topographie (qui peut être remplacée par les définitions de déclinaison dans les sous-niveaux). La `<specification>` peut prendre trois formes :
 1. `[]` une chaîne vide. Cela réinitialisera la définition de la déclinaison magnétique.
 2. `[<value> <units>]` définira une valeur unique de déclinaison magnétique (aussi pour les topographies non datées).
 3. `[<date1> <value1> [<date2> <value2> ...] <units>]` définira la déclinaison magnétique pour plusieurs dates différentes. Ensuite, la déclinaison de chaque visée sera définie en fonction de la date spécifiée par les données de l'objet. Si vous souhaitez définir explicitement la déclinaison pour les données de topographies non datées, utilisez '-' au lieu de la date.

Si aucune déclinaison magnétique n'est spécifiée mais qu'un système de coordonnées géodésique est défini, la déclinaison est automatiquement calculée à l'aide du modèle géomagnétique intégré.

N.B. : La déclinaison magnétique est positive lorsque le nord magnétique est à l'Est du nord vrai.

- `person-rename <ancien nom> <nouveau nom>` ▸ renomme une personne dont le nom a été changé.
- `title <string/blabla>` ▸ description de l'objet.

³ Note : il est impossible de faire référence à un objet rangé parmi les topographies de niveau supérieur.

- **entrance** <nom-de-station> ▸ spécifie l'entrée principale de la cavité représentée par cette topographie. Si elle n'est pas spécifié et qu'il y a au moins une station marquée dans cette topo, celle-ci est également considérée comme une cavité. Cette information est utilisée pour l'exportation dans **cave-list**.

'centreline'

Description : Données du relevé topographique (ligne de cheminement). La syntaxe est empruntée à Survox avec des modifications mineures. Le manuel Survox peut être utile en tant que référence supplémentaire pour l'utilisateur. Un terme synonyme à 'centerline' peut être utilisé.

Syntaxe : **centreline** [OPTIONS]

```

    date <date>
    team <personne> [<roles>]
    explo-date <date>
    explo-team <personne>
    instrument <quantity list> <description>
    calibrate <quantity list> <zero error> [<scale>]
    units <quantity list> [<facteur>] <unites>
    sd <quantity list> <valeur> <unites>
    grade <grade list>
    declination <valeur> <unites>
    grid-angle <valeur> <unites>
    infer <quoi> <on/off>
    mark <type>
    flags <shot flags>
    station <station> <commentaire> [<flags>]
    cs <systeme de coordonnees>
    fix <station> [<x> <y> <z> [<std x> <std y> <std z>]]
    equate <liste de stations>
    data <style> <ordre de lecture>
    break
    group
    endgroup
    walls <auto/on/off>
    vthreshold <nombre> <unites>
    extend <spec> [<station> [<station>]]
    station-names <prefix> <suffix>
    ...
    [SURVEY DaTa]
    ...
endcentreline

```

Contexte : aucun, survey

Options :

- `id <ext_keyword>` ▷ id de l'object
- `author <date> <personne>` ▷ auteur des données et date de création
- `copyright <date> <string>` ▷ date du copyright date et désignation du copyright
- `title <string>` ▷ description de l'object

Command-like options :

- `date <date>` ▷ Date de la topographie (des mesures). Si plusieurs dates sont spécifiées, un intervalle de temps est créé
- `explo-date <date>` ▷ Date d'exploration. Si plusieurs dates sont spécifiées, un intervalle de temps est créé.
- `team <personne> [<roles>]` ▷ un membre de l'équipe de topographie. Le premier argument est son nom, les autres décrivent les rôles de la personne dans l'équipe (facultatif - non utilisé actuellement). Les mots-clés de rôle pris en charge sont les suivants : station, length, tape, [back]compass, [back]bearing, [back]clino, [back]gradient, counter, depth, station, position, notes, pictures, pics, instruments (insts), assistant (dog).
- `explo-team <personne>` ▷ un membre de l'équipe d'exploration.
- `instrument <quantity list> <description>` ▷ description de l'instrument utilisé obtenir les données du relevé topographique (mêmes mots clés que pour le rôle de l'équipier)
- `infer <quoi> <on/off>` ▷ '`infer plumbs on`' indique au programme d'interpréter les gradients $\pm 90^\circ$ comme HAUT / BAS (cela signifie qu'aucune correction n'est appliquée au clinomètre). '`infer equates on`' indique au programme d'interpréter au cas par cas les visées avec une longueur égale à 0 en tant que commandes équivalentes (ce qui signifie qu'aucune correction n'est appliquée au décamètre)
- `declination <valeur> <units>` ▷ définit la déclinaison magnétique pour les visées suivantes

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

La déclinaison est positive lorsque le nord magnétique est à l'est du nord vrai. Si aucune déclinaison n'est spécifiée ou si la déclinaison est réinitialisée (-), une valeur de déclinaison valide est recherchée dans toutes les topographies dans lesquelles se trouve les données de cet objet. Voir les options de déclinaison magnétique de la commande `survey`.

- `grid-angle <valeur> <units>` ▷ spécifie l'angle de la grille magnétique (déclinaison par rapport au nord de la grille).
- `sd <quantity list> <valeur> <units>` ▷ (sd mis pour déviation standard) définit l'écart-type pour les mesures données. La liste quantité peut contenir les mots-clés

suivants : length, tape, bearing, compass, gradient, clino, counter, depth, x, y, z, position, easting, dx, northing, dy, altitude, dz.

Pour utiliser correctement cette commande, vous devez comprendre ce qu'est un écart-type (ou déviation standard en anglais). Il attribue une valeur à la "propagation" des erreurs dans une mesure. En supposant que celles-ci soient normalement distribuées (courbe de Gauss), nous pouvons affirmer que 95,44 % des longueurs réelles se situeront dans les limites de deux écarts-types (2σ) de la longueur mesurée. Prenons l'exemple des mesures de longueur : un écart-type de 0,25 m signifie que la longueur réelle de ces mesures se situe dans la limite de $\pm 0,5$ m de leur valeur enregistrée par l'instrument dans 95,44 % des cas. Donc, si la mesure est de 7,34 m, la longueur réelle sera très probablement comprise entre 6,84 m et 7,84 m. NB : cet exemple correspond à la "classe 3" de la BCRA (British Cave Research Association).

- **grade** *<grade list>* ▷ définit les écarts-types en fonction de la classification spécifique des topographies établie par la BCRA (British Cave Research Association ; voir détails plus loin pour la commande grade). Tous les écarts-type (sd) ou degrés (grade) spécifiés précédemment sont perdus. Si vous souhaitez changer de SD, utilisez l'option **sd** après cette commande. Si plusieurs degrés de précision différents sont spécifiés, seul le dernier s'applique. Vous pouvez spécifier des écarts-types pour toute la topographie ou uniquement pour une position donnée. Si vous souhaitez combiner les deux, vous devez les utiliser dans une seule ligne grade.
- **units** *<quantity list>* [*<facteur>*] *<units>* ▷ définir les unités utilisées pour les mesures données (même contenu de liste quantité que pour sd).
- **calibrate** *<quantity list>* *<zero error>* [*<echelle>*] ▷ calibrate est utilisé pour spécifier les corrections (étalonnage) d'instruments de mesure, via une erreur de zéro et un facteur d'échelle. Par défaut, l'erreur zéro est de 0.0 et le facteur d'échelle de 1.0 pour toutes les quantités mesurées. La valeur mesurée sera ensuite calculée à l'aide de la formule suivante : *valeur mesuree = (valeur lue - zero error) × scale*. Même contenu de liste quantité que pour sd (voir ci-dessus).
- **break** ▷ peut être utilisé avec des données entrelacées pour séparer deux cheminements.
- **mark** [*<station list>*] *<type>* ▷ définit le type de stations nommées. *<type>* *i* est l'un des éléments suivants : fixed (fixe), painted (marquage à la peinture) et temporary (temporaire ; valeur par défaut). S'il n'y a pas de liste de stations, toutes les stations suivantes sont marquées avec le type choisi.
- **flags** *<shot flags>* ▷ définir des marqueurs pour les visées suivantes. Les marqueurs acceptés sont : **surface** (pour les mesures de surface), **duplicate** (pour des visées dupliquées, **splay** (pour les visées latérales d'habillage qui sont masquées sur les dessins et les modèles par défaut). Ceux-ci sont exclus des calculs de longueur.

Toutes les visées dont l'une des stations est marquée soit '.' soit '-' sont définies comme des visées latérales d'habillage par défaut (voir aussi la commande **data**)

Si le marqueur est défini sur `approx[imate]`, la visée est incluse dans les calculs de longueur totale, mais est également affichée séparément dans les statistiques de la topographie. Ce marqueur doit être utilisé pour les visées incorrectes ou imprécises et qui doivent être réexaminées.

De plus, “not” (non) est autorisé avant un marqueur.

- `station <station> <commentaire> [<flags>]` ▷ ajouter le commentaire de la station et ses marqueurs. Si “” est indiqué comme commentaire, il est ignoré.

Les marqueurs acceptés sont : `entrance` (entrée), `continuation`, `air-draught[:winter/summer]` (courant d’air [:hiver / été], `sink` (perte), `spring` (arrivée d’eau), `doline`, `dig` (surcreusement, désobstruction), `arch` (voûte), `overhang` (surplomb). De plus, `not` (non) est autorisé avant un marqueur, pour supprimer le flag ajouté précédemment.

5.3

Vous pouvez également spécifier des attributs personnalisés pour une station à l’aide du flag `attr`, suivi du nom et de la valeur de l’attribut. Exemple:

```
station 4 "puits \'a explorer" continuation attr code "V"
```

S’il existe un secteur qui a été exploré, mais pas encore topographié, la longueur explorée estimée de ce secteur peut être ajoutée à une station avec le flag `continuation`. Ajoutez simplement `explored <explored-length>` aux marqueurs de la station. Les longueurs explorées font partie des statistiques topographiques de la cavité et sont affichées séparément. Exemple:

```
station 40 "ugly crollway" continuation explor\'ee sur 100 m
```

- `cs <coordinate system>` ▷ système de coordonnées pour les stations ayant des coordonnées fixes.
- `fix <station> [<x> <y> <z> [<std x> <std y> <std z>]]` ▷ fixe les coordonnées de la station topo (avec des erreurs spécifiques). Seule la conversion d’unités, et non l’étalonnage, pourra leur être appliquée
- `equate <station list>` ▷ définit des points topos équivalents.
- `data <style> <readings order>` ▷ définit le style des données (normal, topofil, diving, cartesian, cylpolar—coordonnées polaires cylindriques : dans ce cas les mesures de longueurs sont horizontales et non en suivant la pente—, dimensions, nosurvey—pas de topographie—) et l’ordre de lecture. La zone de l’ordre de lecture utilise l’un des mots-clés suivants : station, from, to, tape/length, [back]compass/[back]bearing, [back]clino/[back]gradient, depth, fromdepth, todepth, depthchange, counter, fromcount, tocount, northing, easting, altitude, up/ceiling⁴, down/floor, left, right, ignore, ignoreall.

C’est-à-dire : station, depuis, vers, décamètre/longueur, compas/azimuth et mesure inverse, clinomètre/pente et mesure inverse, profondeur, profondeur depuis, profondeur vers, changement de profondeur, compteur (topofil), compteur depuis, compteur vers,

⁴ La dimension peut être spécifiée comme une paire `<from> <to>` c’est à dire la dimension au début et à la fin de la visée.

vers le nord, vers l'est, altitude, haut/voûte, bas/sol, gauche, droite, ignorer, tout ignorer.

Voir le manuel du logiciel Survex pour plus de détails.

Pour les données entrelacées, les mots-clés de nouvelle ligne et de direction sont pris en charge. Si des mesures avant-arrière (normale / inverse) de compas ou de clinomètre sont enregistrées, la moyenne des deux sera calculée.

Si l'une des stations de visée porte le nom '.' ou '-', le marqueur splay (visée latérale d'habillage) lui est attribué. Le point '.' devrait être utilisé pour les visées se terminant sur des éléments situés à l'intérieur de la galerie topographiée, le tiret '-' pour les visées se terminant sur les parois, au sol ou au plafond. Bien que Therion ne fasse pas encore de distinction entre eux, il devrait être utilisé pour améliorer la modélisation 3D à l'avenir. 5.3

- **group**
- **endgroup** ▷ Cette paire **group/endgroup** permet à l'utilisateur d'apporter des modifications temporaires dans presque tous les paramètres (calibrate, units, sd, data, flags...).
- **walls <auto/on/off>** ▷ Active / désactive la création de la forme de la galerie (parois automatiques) à partir des données LRUD pour les visées suivantes. Si cette option est définie sur **auto**, le secteur n'est créé que s'il n'y a pas de scrap référençant la ligne de cheminement (centerline).
- **vthreshold <nombre> <units>** ▷ seuil pour interpréter les lectures LRUD comme des lectures gauche-droite-avant-arrière perpendiculaires à la visée (habillage).

Si le secteur est horizontal (**inclination < vthreshold**), LR est perpendiculaire à la visée et UD est vertical.

Si le secteur est plus ou moins vertical (**inclination > vthreshold**), même UD devient perpendiculaire au plan - sinon, le rendu ne serait pas correct. Dans le cas de visées verticales, la valeur UD est interprétée comme une dimension nord-sud de la station afin de permettre la modélisation tubulaire des verticales (visualisation des puits)

- **extend <spec> [<station> [<station>]]** ▷ contrôle la direction de la ligne de cheminement. **<spec>** est l'un des mots-clés suivants :

normal/reverse ▷ étendre les stations données et suivantes dans la même direction / ou la direction inverse par rapport à la station précédente. Si deux stations sont indiquées, la direction est appliquée uniquement à la visée donnée.

left/right ▷ comme ci-dessus, mais la direction est spécifiée explicitement.

vertical ▷ ne déplacez pas la station (visée/shot) dans la direction X, , utilisez uniquement la composante Z (verticale) de la visée.

start ▷ spécifie la station de départ (visée).

ignore ▷ ignore la station spécifiée (visée), continuer et prolonger le cheminement avec une autre station (visée) si c'est possible.

hide ▷ masquer (ne pas afficher) la station spécifiée (visée) en élévation prolongée. Si aucune station n'est spécifiée, **<spec>** est valide pour les visées spécifiées suivantes.

- **station-names <prefix> <suffix>** ▷ ajoute un préfixe / suffixe donné à toutes les stations topo de la ligne de cheminement en cours (centerline). Enregistre bon nombre de signes typographiques.

'scrap'

Description : Un scrap est un morceau de topographie 2D qui ne contient pas de parties qui se superposent (tous les passages peuvent être dessinés sur un papier sans se chevaucher). Pour les grottes courtes et simples, la cavité entière peut appartenir à un seul scrap. Dans les systèmes complexes, un scrap représente généralement une salle ou une galerie. Idéalement, un scrap contient environ 100 m de cavité⁵. Chaque scrap est traité séparément par METAPOST; les scraps trop volumineux peuvent dépasser la mémoire de METAPOST et provoquer des erreurs.

Un scrap de topo se compose de symboles, de points, de lignes et de zones. Voir le chapitre *Comment le dessin est-il construit ?* pour une explication sur comment et dans quel ordre ces éléments sont affichés.

La bordure d'un scrap est constituée de lignes avec les options **-outline out** ou **-outline in** (les parois de galerie ont **-outline in** par défaut). Ces lignes ne doivent pas se croiser—sinon Therion (METAPOST) ne peut pas déterminer l'intérieur du scrap et METAPOST envoie un message d'avertissement (Warning) "**scrap outline intersects itself**" (Le contour du scrap se coupe lui-même).

Chaque bloc a son propre système de coordonnées cartésiennes local, qui correspond généralement au quadrillage sur papier millimétré (si vous mesurez manuellement les coordonnées des symboles de la topo) ou aux pixels de l'image numérisée (si vous utilisez XTherion). Therion effectue la transformation de ce système de coordonnées local en coordonnées réelles à l'aide des positions des stations de relevé, qui sont spécifiées à la fois dans le scrap en tant que symboles de points topo et dans les données de la ligne de cheminement (centerline). Si le scrap ne contient pas au moins deux stations de levé avec la référence **-name**, vous devez utiliser l'option **-scale** pour calibrer le scrap (ceci est courant pour les sections transverses).

⁵ Si nécessaire, les scraps peuvent être beaucoup plus petits—pour afficher juste quelques mètres de la cavité. Lorsque vous décidez de la taille du scrap, veuillez prendre en compte les éléments suivants : L'utilisation de petits scraps peut prendre plus de temps au cartographe pour optimiser les jonctions de scraps. D'un autre côté, les algorithmes de déformation de topographie déforment probablement moins les scraps les plus petits. L'utilisation de scraps trop volumineux peut épuiser la mémoire de METAPOST si des remplissages de passages sont fréquemment utilisés, de plus l'éditeur de topos de XTherion est beaucoup moins réactif lors de l'édition de gros morceaux.

La transformation comprend les étapes suivantes :

- Transformation linéaire (décalage, mise à l'échelle et rotation) qui fait correspondre 'au mieux' les stations dessinées du scrap aux stations réelles. 'Au mieux' signifie ici que la somme des distances au carré entre les stations correspondantes avant et après la transformation est minimale. Le résultat est affiché en rouge si l'option `debug` (débogage) de la commande `layout` (présentation / mise en page) est activée (`on`).
- Transformation non linéaire du scrap qui (1) déplace les stations topographiques à la position correcte, (2) est continue. Résultat affiché en bleu en mode `debug` (débogage).
- Transformation non linéaire du scrap qui (1) déplace les points joints, (2) ne déplace pas les stations topographiques, (3) est continue. Enfin, la position des points de contrôle des courbes est ajustée pour préserver la douceur du rendu. Le résultat produit la topographie finale.

Syntaxe : `scrap <id> [OPTIONS]`

```
... commandes point, line (ligne) et area (aire) ...
endscrap [<id>]
```

Contexte : aucun, survey

Arguments :

- `<id>` ▸ identifiant de scrap

Options :

- `projection <specification>` ▸ spécifie la projection du dessin. Chaque projection est identifiée par un type et éventuellement par un index sous la forme `type[:index]`. L'index peut être n'importe quel mot-clé. Les types de projection suivants valides sont les suivants :
 1. `none` ▸ Aucune projection, utilisée pour les sections transverses ou les topographies indépendantes des données du relevé (par exemple, numérisation d'anciennes topos où aucune donnée originelle n'est disponible). Aucun index n'est autorisé pour cette projection.
 2. `plan` ▸ Projection du plan de base (par défaut).
 3. `elevation` ▸ Projection orthogonale (pour une coupe projetée), qui prend éventuellement la direction de la vue comme argument (par exemple `[elevation 10]` ou `[elevation 10 deg]`).
 4. `extended` ▸ Etendue (pour une coupe développée).
- `scale <specification>` ▸ Est utilisé pour une pré-mise à l'échelle (conversion des coordonnées de pixels en mètres) des données du scrap. Si la projection du scrap est nulle, il s'agit de la seule transformation effectuée avec des coordonnées. La `<specification>` peut prendre quatre formes :
 1. `<nombre>` ▸ `<nombre>` mètres par unité de dessin.

2. [`<nombre> <unites de longueur>`] ▷ `<nombre> <unites de longueur>` par unité de dessin.
 3. [`<num1> <num2> <unites de longueur>`] ▷ `<num1>` unités de dessin correspond en réalité à `<num2> <unites de longueur>`.
 4. [`<num1> ... <num8> [<unites de longueur>]`] ▷ il s'agit du format le plus général, dans lequel vous spécifiez, dans l'ordre, les coordonnées x et y de deux points du scrap et de deux points de la réalité. Vous pouvez également spécifier des unités pour les coordonnées des 'points dans la réalité'. Ce formulaire vous permet d'appliquer à la fois la mise à l'échelle et la rotation.
- `cs <systeme de coordonnees>` ▷ suppose que les coordonnées locales (calibrées) des scraps sont données dans le système de coordonnées spécifié. C'est utile pour le placement absolu des esquisses importées lorsqu'aucune station topo n'est spécifiée.⁶
 - `stations <liste de noms de stations>` ▷ stations que vous souhaitez tracer sur le scrap, mais qui ne sont pas utilisées pour la transformation de celui-ci. Vous n'avez pas à les spécifier (dessiner) avec la commande `point station`.
 - `sketch <nom_de_fichier> <x> <y>` ▷ spécification de repérage bitmap d'une esquisse sous-jacente (coordonnées de son coin inférieur gauche).
 - `walls <on/off/auto>` ▷ spécifie si le scrap doit être utilisée dans la reconstruction de modèle 3D.
 - `flip (none)/horizontal/vertical` ▷ (re)tourne le scrap après transformation.
 - `station-names <prefix> <suffix>` ▷ ajoute un préfixe / suffixe donné à toutes les stations topo du scrap actuel. Enregistre bon nombre de signes typographiques.
 - `author <date> <personne>` ▷ auteur des données (scrap) et date de création.
 - `copyright <date> <string>` ▷ date et nom du copyright
 - `title <string>` ▷ description de l'objet.

'point'

Description : Point est une commande permettant de dessiner un symbole de point sur la topographie.

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

Contexte : scrap

⁶ S'il y a des stations topo dans le scrap, la spécification `cs` est donc ignorée.

Arguments :

- `<x>` et `<y>` sont les coordonnées d'un objet.
- `<type>` détermine le type d'objet. Les termes suivants sont acceptés :

Objets spéciaux : `station`⁷, `section`⁸, `dimensions`⁹ ; (idem en français)

Etiquettes (labels) : `label`, `remark`, `altitude`¹⁰, `height`¹¹, `passage-height`¹², `station-name`¹³, `date` ; (etiquette, remarque, altitude, hauteur, hauteur de galerie, nom de station, date) ;

*Symboles de remplissages des passages :*¹⁴ `bedrock`, `sand`, `raft`, `clay`, `pebbles`, `debris`, `blocks`, `water`, `ice`, `guano`, `snow` ; (roche mère ou en place ou substrat rocheux, sable, gour, argile, cailloux, débris, blocs, eau, glace, guano, neige) ;

speleothèmes: `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` ; (coulée de calcite, lait de lune, stalactite, stalagmite, pilier, draperie, excentrique, fistuleuse, cristal, paroi concrétionnée, choux-fleurs, disque, gypse, fleur de gypse, aragonite, pisolithe, bassin de gour, bordure de gour, anastomose, karren/karst (rainures de dissolution karstique), coups de gouge, flûte, cne de gour, argile) ; 5.4

Équipement: `anchor`, `rope`, `fixed-ladder`, `rope-ladder`, `steps`, `bridge`, `traverse`, `camp`, `no-equipment` ; (amarrage, corde, échelle fixe, échelle de corde, marches, pont, traverse, campement, pas d'équipement) ;

Fin de galerie : `continuation`, `narrow-end`, `low-end`, `flowstone-choke`, `breakdown-choke`, `clay-choke`, `entrance` ; (continuation, extrémité étroite, extrémité basse, bouchon de calcite, étroiture infranchissable, bouchon d'argile, entrée) 5.4

⁷ Station topo. Pour chaque scrap (à l'exception des scraps sans projection - avec option 'none'), au moins une station ayant une référence de station (option `-name`) doit être spécifiée.

⁸ `section` est une ancre permettant de placer une section transverse à cet endroit. Ce symbole n'a aucune représentation visuelle. La section transverse doit être dans un scrap séparé avec la projection 'none' spécifiée. Vous pouvez le spécifier via l'option `-scrap`.

⁹ Utilisez l'option `-value` pour spécifier les dimensions de la galerie au-dessus / au-dessous du plan de la ligne médiane utilisé lors de la création d'un modèle 3D.

¹⁰ Etiquette d'altitude générale. Toutes les altitudes sont exportées sous la forme d'une différence par rapport à l'origine de la grille Z (0 par défaut). Pour afficher l'altitude sur la paroi de la galerie, utilisez l'option `altitude` pour n'importe quel point constituant la ligne de cette paroi.

¹¹ Hauteur des formations à l'intérieur du secteur topographié (comme un puits, un cne d'absorption, etc.); voir plus loin pour plus de détails.

¹² Hauteur de la galerie ou de la salle; voir plus loin pour plus de détails.

¹³ Si aucun texte n'est spécifié, le nom de la station la plus proche est utilisé.

¹⁴ Contrairement aux autres symboles ponctuels, ceux-ci sont découpés par la bordure. Voir le chapitre *Comment la topographie est-elle construite ?*

Autres: `dig`, `archeo-material`, `paleo-material`, `vegetable-debris`, `root`, `water-flow`, `spring`¹⁵, `sink`, `ice-stalactite`, `ice-stalagmite`, `ice-pillar`, `gradient`, `air-draught`¹⁶, `map-connection`¹⁷, `extra`¹⁸, `u`¹⁹ ; (creusement, matériel archéologique, matériel paléontologique, débris végétaux, racines, écoulement d'eau, source, perte, stalactite de glace, stalagmite de glace, pilier de glace, gradient, courant d'air, connexion topo, extra, utilisateur).

Options :

- `subtype <mot-clef>` ▷ détermine le sous-type de l'objet. Pour chaque type donné, voici les sous-types possibles :

`station`:²⁰ (c'est le sous-type décrivant une station topo) `temporary` (temporaire ; par défaut), `painted` (marquage à la peinture), `natural` (élément naturel), `fixed` (fixe) ;

`air-draught` (courant d'air) : `winter` (hiver), `summer` (été), `undefined` (indéfini ; par défaut) ;

`water-flow` : `permanent` (par défaut), `intermittent`, `paleo`.

Le sous-type peut également être indiqué directement dans la spécification `<type>` en utilisant les deux points superposés `‘.’` comme séparateur.²¹

Toute spécification de sous-type peut être utilisée avec un type défini par l'utilisateur (`u`). Dans ce cas, vous devrez également définir le symbole METAPOST correspondant (voir le chapitre *Nouveaux symboles topographiques*).

- `orientation/orient <nombre>` ▷ définit l'orientation du symbole. Si rien n'est spécifié, il est orienté vers le nord. $0 \leq \text{number} < 360$.
- `align` ▷ alignement du symbole ou du texte. Les valeurs suivantes sont acceptées : `center`, `c`, `top` (haut), `t`, `bottom` (bas), `b`, `left` (gauche), `l`, `right` (droite), `r`, `top-left`, `tl`, `top-right`, `tr`, `bottom-left`, `bl`, `bottom-right`, `br` (aller à la ligne).
- `scale` ▷ échelle (taille) du symbole, qui peut être : `tiny` (`xs`), `small` (`s`), `normal` (`m`), `large` (`l`), `huge` (`xl`) ou une valeur numérique. 'normal' est la valeur par défaut. Les tailles augmentent d'un facteur $\sqrt{2}$ donc si $xs \equiv 0.5$, $s \equiv 0.707$, $m \equiv 1.0$, $l \equiv 1.414$ et $xl \equiv 2.0$.
- `place <bottom/default/top>` ▷ change l'ordre d'affichage sur la topographie finale.

¹⁵ Utilisez toujours les symboles `spring` (arrivée d'eau) et `sink` (perte) avec une flèche `water-flow` pour le débit d'eau.

¹⁶ Le nombre de barbules est défini en fonction de l'option `-scale`.

¹⁷ Point virtuel, utilisé pour indiquer la connexion entre des topos déplacées (coupe développée, décalage de topographie).

¹⁸ Ajout de points au morphing.

¹⁹ Pour définir des points de symboles par utilisateur.

²⁰ Si le sous-type de station n'est pas spécifié, Therion le lit à partir de la ligne de cheminement, s'il est spécifié `water-flow` (arrivée d'eau) le sous-type sera : `permanent` (par défaut), `intermittent` ou `paleo`.

²¹ Par exemple `station:fixed`

- `clip <on/off>` ▸ spécifie si un symbole est coupé par la bordure du scrap. Vous ne pouvez pas spécifier cette option pour les symboles suivants : `station`, `station-name`, `label`, `remark`, `date`, `altitude`, `height`, `passage-height`.
- `dist <distance>` ▸ valide pour des points supplémentaires, spécifie la distance jusqu'à la station la plus proche (ou à la station spécifiée à l'aide de l'option `-from`). Si non spécifié, la valeur appropriée des données LRUD est utilisée.
- `from <station>` ▸ valable pour des points supplémentaires, spécifie la station de référence.
- `visibility <on/off>` ▸ montre/cache le symbole.
- `context <point/line/area> <symbol-type>` ▸ (doit être utilisé avec les options `symbol-hide` et `symbol-show` de la mise en page (layout)) le symbole sera caché/affiché conformément aux règles spécifiées par `<symbol-type>`.²²
- `id <ext_keyword>` ▸ identifiant de symbole.

Options spécifiques de type de point :

- `name <reference>` ▸ si le type de point est `station`, cette option donne la référence à la station topo réelle.
- `extend [prev[ious] <station>]` ▸ si le type de point est `station` et que la projection du scrap est en coupe développée, vous pouvez ajuster l'extension de l'axe de la ligne de cheminement à l'aide de cette option.
- `scrap <reference>` ▸ si le type de point est `section`, il s'agit d'une référence à une section transverse du scrap.
- `explored <length>` ▸ Si le type de point est `continuation`, vous pouvez spécifier la longueur des zones explorées mais non encore topographiées. Cette valeur est ensuite affichée dans les statistiques `survey/cave`.
- `text` ▸ texte d'une étiquette, d'une remarque ou d'une suite de galerie. Il peut contenir les mots-clés suivants pour le formatage :²³

`
` ▸ saut de ligne (break).

`<center>/<centre>`, `<left>`, `<right>` ▸ alignement des lignes pour les étiquettes multilignes. Ignoré s'il n'y a pas de balise `
`.

`<thsp>` ▸ petit espace.

²² Exemple : si vous spécifiez `-context point air-draught` sur une étiquette indiquant la date d'observation, la commande `symbol-hide point air-draught` masquera à la fois la flèche du courant d'air et l'étiquette correspondante.

²³ Pour un export en SVG, seuls les mots-clés `
`, `<thsp>`, `<it>`, `<bf>`, `<rm>` et `<lang:XX>` sont pris en compte; tous les autres sont ignorés.

`<rm>`, `<it>`, `<bf>`, `<ss>`, `<si>` ▸ commutateurs de police de caractères (normal, italique, gras, normal simple, simple italique).

`<rtl>` and `</rtl>` ▸ marque le début et la fin d'un texte écrit de droite à gauche. 5.3

`<lang:XX>` ▸ crée une étiquette multilingue (voir le type string (chaîne) pour une description détaillée). 5.3

- **value** ▸ valeur de la hauteur, hauteur d'une zone, étiquette d'altitude ou dimensions à un certain point.

Hauteur : selon le signe de cette valeur (positif, négatif ou sans signe), ce type de symbole représente en général la hauteur d'une cheminée, la profondeur d'un trou ou la hauteur des ressauts. La valeur numérique peut éventuellement être suivie de '?' si elle est estimée. Des unités peuvent être ajoutées (par exemple, `-value [40? ft]`).

hauteur du passage : les quatre formes de valeur suivantes sont acceptés : `+<nombre>` (la hauteur du plafond), `-<nombre>` (la profondeur du sol vers le bas ou de la profondeur de l'eau), `<nombre>` (la distance entre le plancher et la voûte) et `[+<nombre> -<number>]` (la distance au plafond et la distance au sol).

Dimensions: `-value [<above> <below> [<units>]]` spécifie les dimensions de la zone au-dessus et au-dessous du plan d'axe utilisé dans le modèle 3D.

'line'

Description : Line est une commande permettant de dessiner une ligne sur la topographie. Chaque ligne est orientée et son aspect final peut dépendre de son orientation (par exemple, le sens des coches d'une bordure de pente). La règle générale est que l'espace libre (le vide) est à gauche et la roche à droite. Exemples : le bas d'une pente, le haut d'une cheminée et l'intérieur d'une galerie se trouvent respectivement à gauche des symboles de pente, de cheminée ou de paroi.

Syntaxe : `line <type> [OPTIONS]`

```
[OPTIONS]
...
[LINE DaTa]
...
[OPTIONS]
...
[LINE DaTa]
...
endl
```

Contexte : scrap

Arguments :

- **<type>** est un mot-clé qui détermine le type de ligne. Les types suivants sont acceptés :

passages/galleries : **wall**, **contour**, **slope**²⁴, **floor-step**, **pit**, **ceiling-step**, **chimney**, **overhang**, **ceiling-meander**, **floor-meander** ; (paroi, contour, pente, ressaut, puits, décrochement de voûte, cheminée, surplomb, méandre de plafond, méandre au sol) ;

Remplissage des passages : **flowstone**, **moonmilk**, **rock-border**²⁵, **rock-edge**²⁶, **water-flow** ; (plancher stalagmitique, lait de lune, bordure rocheuse, arête rocheuse, écoulement d'eau) ;

Labels / étiquettes: **label** ; (étiquette) ;

Spécial: **border**, **arrow**, **section**²⁷, **survey**²⁸, **map-connection**²⁹, **u**³⁰.

Command-like options :

- **subtype <keyword>** ▸ détermine le sous-type de ligne. Les sous-types suivants sont valides pour les types donnés :

wall: **invisible**, **bedrock** (par défaut), **sand**, **clay**, **pebbles**, **debris**, **blocks**, **ice**, **underlying**, **overlying**, **unsurveyed**, **presumed**, **pit**³¹, **flowstone**, **moonmilk** ; (invisible, substrat rocheux, sable, argile, cailloux, débris, blocs, glace, sous-jacente, sus-jacente, non topographiée, présumée, puits, plancher stalagmitique, lait de lune) ;

5.4

border : **visible** (default), **invisible**, **temporary**, **presumed** ; (visible (par défaut), invisible, temporaire, présumée) ;

water-flow : **permanent** (par défaut), **conjectural**, **intermittent**;

survey : **cave** (par défaut), **surface** (par défaut si la ligne de cheminement possède un marqueur de surface)

²⁴ La ligne de pente marque le bord supérieur de la zone en pente. Il est nécessaire de spécifier **l-size** sur au moins un point. La longueur et l'orientation des lignes de pente sont une moyenne des tailles **l-size** spécifiées et des orientations aux points les plus proches. S'il n'y a pas de spécification d'orientation, les marques de pente sont perpendiculaires à la ligne de pente.

²⁵ Contour extérieur de gros rochers. Si la ligne est fermée, elle est remplie avec la couleur d'arrière-plan.

²⁶ Arêtes intérieures de gros rochers.

²⁷ Ligne indiquant la position de la section transversale. Si les deux points de contrôle (points rouges) d'une courbe de Bézier (ligne grise) sont indiqués, la ligne de section (bleue) est dessinée vers la projection perpendiculaire (en pointillé) du premier point de contrôle et à partir de la projection (en pointillé) du second point de contrôle de cette section. Aucune courbe de section n'est affichée.

5.3



²⁸ La ligne de cheminement du levé topographique est automatiquement dessinée par Therion.

²⁹ Utilisé pour indiquer une connexion entre des topographies (en décalage, ou les mêmes points en coupe développée).

³⁰ Utilisé pour les symboles de lignes définis par l'utilisateur.

³¹ Habituellement ouvert à la surface.

Le sous-type peut également être mentionné directement dans la spécification `<type>` en utilisant ‘:’ comme séparateur.³²

Toute indication de sous-type peut être utilisée avec un type défini par l'utilisateur (`u`). Dans ce cas, vous devrez également définir le symbole METAPOSTcorrespondant (voir le chapitre *Nouveaux symboles topographiques*).

- `[LINE DaTa]` spécifie soit les coordonnées d'un segment de ligne `<x> <y>`, ou les coordonnées d'une courbe de Bézier `<c1x> <c1y> <c2x> <c2y> <x> <y>`, où `c` indique le point de contrôle.
- `close <on/off/auto>` ▸ détermine si une ligne est fermée ou non.
- `mark <mot_clef>` ▸ est utilisé pour marquer le point sur la ligne (voir la commande `join`).
- `orientation/orient <nombre>` ▸ orientation des symboles sur la ligne. S'il n'est pas spécifié, il est perpendiculaire à la droite et orienté du côté gauche. $0 \leq \text{number} < 360$.
- `outline <in/out/none>` ▸ détermine si la ligne sert de limite à un scrap. La valeur par défaut est ‘out’ pour les parois, ‘none’ pour les autres lignes. Utiliser `-outline in` pour de gros pilers etc.
- `reverse <on/off>` ▸ si les points sont indiqués dans l'ordre inverse.
- `size <nombre>` ▸ largeur de trait (les tailles gauche et droite prendront la moitié de cette valeur).
- `r-size <nombre>` ▸ taille de la ligne à droite.
- `l-size <nombre>` ▸ même chose mais à gauche. Utilisé pour le type `slope` (pente).
- `smooth <on/off/auto>` ▸ si la ligne doit être lissée au point donné. `Auto` est la valeur par défaut.
- `adjust <horizontal/vertical>` ▸ décale le point de la ligne à aligner horizontalement / verticalement avec le point précédent (ou le point suivant s'il n'y a pas de point précédent). Le résultat est un segment de ligne horizontale / verticale). Si tous les points de la ligne ont cette option, ils sont alignés respectivement sur la coordonnée moyenne *y* ou *x*. Cette option n'est pas autorisée dans la projection `plan`.
- `place <bottom/default/top>` ▸ changements de la disposition lors de l'affichage sur la topographie finale.
- `clip <on/off>` ▸ spécifie si un symbole est coupé par la bordure d'un scrap.
- `visibility <on/off>` ▸ affiche / cache le symbole.
- `context <point/line/area> <symbol-type>` ▸ (à utiliser avec les options de mise en page (layout) `symbol-hide` et `symbol-show`). Le symbole sera masqué/affiché conformément aux règles du `<symbol-type>` spécifié.

³² E.g. `border:invisible`

Options spécifiques :

- **altitude** *<value>* ▷ ne peut être spécifié qu’avec le type paroi (**wall**). Cette option crée une étiquette d’altitude sur la paroi. Toutes les altitudes sont exportées sous la forme d’une différence par rapport à l’origine de la grille *Z* (0 par défaut). Si la valeur est spécifiée, elle donne la différence d’altitude du point sur la paroi par rapport à la station la plus proche. La valeur peut être préfixée par le mot-clé “**fix**”, alors aucune station proche ne sera prise en compte ; la valeur absolue donnée étant utilisée à la place. Les unités peuvent suivre la valeur. Exemples : **+4**, **[+4 m]**, **[fix 1510 m]**.
- **border** *<on/off>* ▷ cette option ne peut être spécifiée qu’avec le type de symbole ‘slope’ (pente). Il active / désactive la ligne de limite de pente.
- **direction** *<begin/end/both/none/point>* ▷ doit être utilisé uniquement avec le type ‘section’. Il indique où placer une flèche de direction sur la ligne de section transversale. ‘none’ est la valeur par défaut.
- **gradient** *<none/center/point>* ▷ Ne peut être utilisé qu’avec le type **contour** ; indique où ajouter une marque de gradient (pente) sur la ligne **contour**. S’il n’y a pas de spécification **gradient**, le comportement est dépendant de symbol-set (i.e. pas de tick avec l’UIS, tick au milieu avec SKBB).
- **head** *<begin/end/both/none>* ▷ peut être utilisé uniquement avec le type ‘arrow’ (flèche) et indique où placer la pointe de la flèche. ‘end’ est la valeur par défaut.
- **text** *<string>* ▷ valable uniquement pour les lignes d’étiquettes (label).
- **height** *<value>* ▷ hauteur d’un puits ou d’une paroi ; disponible comme variable numérique `METAPOSTaTTR_height`. 5.4

Options :

- **id** *<ext_keyword>* ▷ Identifiant du symbol.

‘area’

Description : Une zone est spécifiée par les lignes de bordures environnantes. Elles peuvent être de n’importe quel type, mais doivent être répertoriées dans l’ordre et chaque paire de lignes consécutives doit se croiser. Pour vous assurer que les lignes se coupent même après la transformation, vous pouvez, par exemple, continuer une bordure de lac 1 cm derrière une paroi de la galerie—ces chevauchements seront automatiquement découpés par une des bordures du scrap. Pour y parvenir, vous pouvez utiliser une bordure invisible à l’intérieur de la galerie.

Syntaxe : **area** *<type>*
 place *<bottom/default/top>*
 clip *<on/off>*
 visibility *<on/off>*
 ... **border line references** ...

endarea

Contexte : scrap

Arguments :

- **<type>** est un des termes suivants : **water**, **sump**, **sand**, **debris**, **blocks**, **flowstone**, **moonmilk**, **snow**, **ice**, **clay**, **pebbles**, **bedrock**³³, **u**³⁴ ; (eau, siphon, sable, débris, blocs, coulée de calcite, moonmilk, neige, argile, galets, roche mère).

Command-like options :

- La données des lignes consistent en leur références de lignes de bord (IDs)
- **place** **<bottom/default/top>** ▸ changements de la disposition lors de l’affichage sur la topo finale.
- **clip** **<on/off>** ▸ spécifie si un symbole est coupé par la bordure du scrap.
- **visibility** **<on/off>** ▸ affiche / cache le symbole.
- **context** **<point/line/area>** **<symbol-type>** ▸ (à utiliser avec les options de mise en page (layout) **symbol-hide** et **symbol-show**). Le symbole sera masqué/affiché conformément aux règles du **<symbol-type>** spécifié.

Options :

- **id** **<ext_keyword>** ▸ Identifiant du symbole.

‘join’

Description : La jonction fonctionne selon deux modes : elle peut relier deux scraps, mais aussi deux ou plusieurs points ou lignes d’une même topographie.

Lorsque vous joignez plus de deux points ou lignes, utilisez une seule commande de jonction pour chacun d’eux, et non une séquence de commandes de jonction pour des paires de points.³⁵

Lorsque vous joignez des scraps, seules les parois sont jointes. Il est préférable de placer une jonction aussi simple que possible dans la galerie, sinon vous devrez spécifier une jonction pour chaque paire d’objets à joindre.³⁶

Lorsque vous voulez joindre plus de deux scraps à la même limite de scrap, une jonction manuelle doit être effectuée, les points de connexion devant être entrés dans une déclaration de jonction.

³³ une aire vide qui peut être utilisée pour nettoyer l’arrière-plan.

³⁴ Pour les symboles définis par l’utilisateur, ils peuvent être suivi d’un sous-type.

³⁵ par exemple : utiliser **join a b c**, et non **join a b** suivi par **join b c**.

³⁶ Si vous voulez qu’un objet coupé par une limite de scrap continue jusqu’à un scrap voisin, utilisez l’option **-clip off** pour cet objet.

When joining more than two scraps at the same scrap border, a manual join must be performed where the connection points must be entered in one join statement.³⁷

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

Contexte : aucun, scrap, survey

Arguments :

- `<pointX>` peut être un ID d'un symbole point ou ligne, optionnellement suivi par un point de ligne spécifique `<id>:<mark>` (e.g. `podangl_131@podangl:mark1`). `<mark>` peut aussi être `'end'` (fin de ligne) ou un point d'index de ligne (où 0 est le premier point).

Il y a un cas spécial lorsque `<point1>` et `<point2>` sont des ID de scraps—en ce cas, les extrémités des scraps les plus proches sont jointes.

Options :

- `smooth <on/off>` indique si deux lignes doivent être connectées en arrondissant l'angle.
- `count <N>` (avec l'utilisation de scraps) > Therion va essayer de joindre les scraps en connectant `N` passages/galleries.

'equate'

Description : Définit l'équivalence des stations topographiques.

Syntaxe : `equate <liste de stations>`

Contexte : aucun, survey

'map'

Description : Une 'map' (topographie) est un assemblage de scraps ou de plusieurs topographies présentant le même type de projection. Il est possible d'inclure un relevé topographique à la topographie : cela affichera la ligne de cheminement sur le dessin. L'objet 'map' simplifie la gestion des données lors de la sélection des données à imprimer. Voir le chapitre *Comment la topographie est-elle construite ?* pour une explication plus détaillée.

Syntaxe : `map <id> [OPTIONS]`

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

³⁷ Comme par exemple `join origScrapLineWest:end upperScrapLineWest:0 lowerScrapLineWest:0` et une autre ligne de commande similaire pour les trois lignes du mur cté Est.

endmap

Contexte : aucun, survey

Arguments :

- `<id>` ▸ identifiant de scraps

Command-like options :

- les lignes de données sont constituées de références de scraps ou de topographies. Notez que vous ne pouvez pas les mélanger.
- Si vous vous référez à la topographie, vous pouvez spécifier le décalage auquel cette sous-topographie sera affichée, ainsi que le type de pré-visualisation de sa position d'origine. La syntaxe est la suivante :

`<map reference> [<offset X> <offset Y> <units>] <above/below/none>`

- les scraps situés après la pause (`break`) seront placés à un autre niveau.
- `preview <above/below> <other map id>` mettra le contour de l'autre topographie en tant qu'aperçu par rapport à la topographie actuelle.

L'aperçu ne s'affiche que si la topo est au niveau (`map-level`) correspondant à celui spécifié par la commande de sélection `select`.

Utilisez la commande `revise` si vous souhaitez ajouter des topographies de niveaux supérieurs à l'aperçu.

- `colo[u]r <couleur>` ▸ définit la couleur de la topographie ; cette option annule le choix automatique lorsque la mise en page (`layout`) spécifie `colour map-fg [map]`.

Options :

- `projection/proj <plan/elevation/extended/none>` ▸ est requis si la topographie contient le relevé.
- `title <string>` ▸ description de l'objet.

- 5.4 • `survey <id>` ▸ associe un relevé à une topographie (par exemple, toutes les statistiques de ce relevé seront utilisées lorsque cette topographie sera sélectionnée en sortie).

'surface'

Description : Spécification de surface (zone ou espace de terrain ou de sol). Il est possible de l'afficher de deux manières : sous forme de topographie numérisée (possible en topo 2D et aussi en modèle 3D)³⁸) ou sous la forme d'une grille de surface de type modèle numérique d'élévation (MNE/MNT/DEM) (en 3D uniquement).

Syntaxe : `surface [<name>]`

³⁸ Vous devrez saisir des données d'altitude pour afficher la carte topographique dans un modèle 3D. Actuellement, seules les cartes JPEG sont prises en charge en 3D.

```

cs <systeme de coordonnees>
bitmap <nom_de_fichier> <calibration>
grid-units <unites>
grid <origin x> <origin y> <x spacing> <y spacing> <x count> <y count>
grid-flip (none)/vertical/horizontal
[grid data]
endsurface

```

Contexte : aucun, survey

Command-like options :

- **cs <systeme de coordonnees>** ▷ système de coordonnées pour le calibrage bitmap et la spécification d'origine de la grille.
- **bitmap <filename> <calibration>** ▷ topographie numérisée.

calibration peut prendre deux formes :

1. **[X1 Y1 x1 y1 X2 Y2 x2 y2 [units]]**, où les variables majuscules *X* / *Y* sont les coordonnées sur l'image (pixels; le coin inférieur gauche valant 0 0), les variables minuscules *x* / *y* sont les coordonnées réelles. Les unités facultatives s'appliquent aux coordonnées réelles (mètres par défaut).
2. **[X1 Y1 station1 X2 Y2 station2]**, où les variables *X* / *Y* en majuscules sont les coordonnées de l'image et les **stations1** et **station2** sont les noms des stations de topographie.

- **grid-units <units>** ▷ unités dans lesquelles la grille est spécifiée. Mètres par défaut.
- **grid <origin x> <origin y> <x spacing> <y spacing> <x count> <y count>**

<origin x> <origin y> ▷ spécifie les coordonnées du coin en bas à gauche (S-W) de la grille

<x spacing> <y spacing> ▷ distance entre les nœuds (intersections) de la grille dans les directions E-W et N-S.

<x count> <y count> ▷ nombre de nœuds (pixels) dans la ligne et nombre de lignes qui forment la grille (voir ci-dessous).

- **[grid data]** ▷ un flux de nombres donnant l'altitude au niveau de la mer des nœuds de la grille. Il commence à l'origine de la grille et remplit la grille en lignes (depuis la ligne W à E ; puis des lignes S à N).
- **grid-flip (none)/vertical/horizontal** ▷ utile si votre grille (exportée depuis un autre programme) doit être retournée.

'import'

Description : Lecture les données du relevé topographique dans différents formats (pour le moment la ligne de cheminement est traitée aux formats *.3d, *.plt, *.xyz). Les stations topographiques peuvent être référencées dans des scraps, etc. Lors de l'importation d'un fichier Survex 3D, les stations sont insérées dans la hiérarchie de la topographie s'il existe une hiérarchie identique à celle du fichier 3D.

Syntaxe : `import <nom_de_fichier> [OPTIONS]`

Contexte : survey / all³⁹

Options :

- `filter <prefix>` ▷ si spécifié, seules les stations avec un préfixe donné et des visées entre elles seront importées. Le préfixe sera supprimé des noms de station.
- `surveys (create)/use/ignore` ▷ spécifie comment importer une structure de relevé topographique (fonctionne uniquement avec les fichiers .3d).
 - `create` ▷ diviser les stations en sous-niveaux. S'il n'y en a pas, les créer.
 - `use` ▷ répartir les stations en sous-niveaux.
 - `ignore` ▷ ne pas répartir les stations en sous-niveaux.
- `cs <systeme de coordonnees system>` ▷ système de coordonnées pour les stations avec des coordonnées fixes.
- `calibrate [<x> <y> <z> <X> <Y> <Z>]` ▷ les coordonnées du fichier importé sont transférées des coordonnées minuscules aux coordonnées majuscules.

³⁹ Uniquement avec les fichiers .3D, où la structure du relevé est spécifiée.

‘grade’

Description : Cette commande est utilisée pour stocker des précisions sur les données prédéfinies pour la ligne de cheminement. Les grades ou degrés intégrés sont : BCRA⁴⁰ et UISv1⁴¹.

Voir la description de l’option `sd` pour la commande `centreline` afin de définir vos propres degrés de précision.

Syntaxe : : `grade <id>`

```
...
[<quantity list> <value> <units>]
...
endgrade
```

Contexte : all

‘revise’

Description : Cette commande est utilisée pour définir ou modifier les propriétés d’un objet déjà existant.

Syntaxe : Pour les objets créés avec des commandes “single line” la syntaxe est la suivante

⁴⁰ voir <http://bcra.org.uk/surveying/> ; la syntaxe est : `BCRan`, où *n* peut être 3 ou 5. Pour info, voici le classement BCRA des topographies suivant leur précision :

Degré 1 » Esquisse de faible précision où aucune mesure n’a été faite

Degré 2 » Peut être utilisée, si nécessaire, pour décrire un croquis dont la précision est intermédiaire entre les niveaux 1 et 3 (à utiliser uniquement si nécessaire).

Degré 3 » Relevé magnétique approximatif. Angles horizontaux et verticaux mesurés à $\pm 2,5^\circ$; distances mesurées à ± 50 cm ; erreur de position de la station inférieure à 50 cm.

Degré 4 » Peut être utilisée, si nécessaire, pour décrire une topographie qui ne répond pas à toutes les exigences de la classe 5 mais est plus précise qu’une topo de classe 3. (à utiliser uniquement si nécessaire).

Degré 5 » Relevé magnétique. Angles horizontaux et verticaux mesurés à $\pm 1^\circ$; les distances doivent être observées et enregistrées au centimètre près et les positions des stations identifiées à moins de 10 cm.

Degré 6 » Relevé magnétique plus précis que la classe 5. C’est-à-dire avec une précision angulaire de $\pm 0,5^\circ$; les lectures du clinomètre doivent avoir la même précision. L’erreur de position de la station doit être inférieure à $\pm 2,5$ cm, ce qui nécessitera l’utilisation de trépieds dans toutes les stations ou d’autres repères de stations fixes.

Degré X » Topographie basée principalement sur l’utilisation d’un théodolite / station totale (tachéomètre).

5.4 ⁴¹ voir <http://www.uisic.uis-speleo.org/UISmappingGrades.pdf> ; la syntaxe est : `UISv1-n`, où *n* est de -1 à 6 ou X ; alors que -1 to 2 sont uniquement là à titre d’information, X nécessite des données sd dans la `centerline`. Les degrés 2 et 4 ne doivent être utilisées que lorsque des conditions matérielles ont empêché la topographie de satisfaire à toutes les exigences requises pour le degré supérieur et qu’il est difficile de recommencer.

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

Pour les objets créés avec des commandes “multi-lignes”, la syntaxe est la suivante :

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

```
...
optionX valueX
data
...
endrevise
```

Contexte : all

Arguments :

L’identifiant signifie ici identifiant de l’objet (car l’identifiant d’un objet que vous voulez réviser doit toujours être spécifié).

Attributs personnalisés

Les objets *survey*, *centreline*, *scrap*, *point*, *line*, *area*, *map* et *surface* peuvent contenir des attributs définis par l’utilisateur sous la forme `-attr <nom> <valeur>`. `<nom>` peut contenir des caractères alphanumériques, `<valeur>` est une chaîne de caractères (string).

Les attributs personnalisés sont utilisés dans l’exportation de la topographie en fonction du format de sortie:

- lors de l’exportation de *shapefiles* (fichiers de données vectorielles pour système d’information géographique—SIG), elles sont écrites directement dans le fichier dbf associé.
- dans les topographies (‘maps’) générées à l’aide de METAPOST(PDF, SVG), les attributs sont écrits dans le fichier source METAPOST sous forme de chaînes (nommées ainsi: `aTTR_<nom>`)) et peuvent être évalués et utilisés par l’utilisateur dans des macros de définition de symboles.

Vous pouvez tester la présence d’une telle variable en utilisant `iif known aTTR_<name>: ... fi.`

XTherion

XTherion est une GUI (interface utilisateur graphique) pour Therion. Il aide beaucoup à la création de fichiers de données d’entrée. Actuellement, il fonctionne dans trois modes principaux : éditeur de texte, éditeur de dessin et compilateur⁴²

⁴² Ici, nous sommes concernés par la création de données, ce qui explique que cette section ne décrit que les deux premiers modes. Pour tout ce qui concerne les actions de compilation, voir le chapitre dédié *Compilation des données*.

XTherion n'est pas nécessaire pour Therion lui-même. Vous pouvez modifier les fichiers d'entrée dans votre éditeur de texte favori et exécuter Therion à partir de la ligne de commande. XTherion n'est donc pas la seule interface graphique pouvant être utilisée avec Therion. Il serait possible d'en écrire une meilleure, ce qui serait plus convivial, plus WYSIWYG, plus rapide, plus robuste et plus facile à utiliser. Des volontaires ?

Ce manuel ne décrit pas des éléments familiers tels que 'si vous souhaitez enregistrer un fichier, allez au menu Fichier et sélectionnez Enregistrer ou appuyez sur Ctrl-s'. Parcourez le menu du haut pendant une minute pour vous familiariser avec XTherion.

Pour chaque mode de fonctionnement, il existe un menu supplémentaire à droite ou à gauche. Les sous-menus peuvent être repliés ; vous pouvez les dérouler en cliquant sur le bouton du menu. Pour la plupart des menus et des boutons, il y a une courte description (traduite) dans la ligne d'état, il n'est donc pas difficile de deviner la signification de chacun des sous-menus. L'ordre des sous-menus sur le cté peut être personnalisé par l'utilisateur. Cliquez avec le bouton droit sur le bouton de menu et sélectionnez dans le menu celui des autres menus avec lequel il doit être échangé.

XTherion—text editor

L'éditeur de texte de XTherion' offre des outils intéressant qui peuvent aider à la création des fichiers texte d'entrée : support de l'encodage Unicode et capacité à ouvrir plusieurs fichiers en parallèle⁴³

Pour faciliter la saisie des données, l'éditeur prend en charge le formatage des données de la ligne de cheminement sous forme de tableau. Il existe un menu Table de données (*Data table*) pour la saisie des données. Il peut être personnalisé en fonction des données de commandes utilisateur en appuyant sur une touche 'Numériser le format des données' (*Scan data format*) lorsque le curseur se trouve sous la spécification de la donnée de commande (option 'data' dans la commande 'centreligne').

XTherion—map editor

L'éditeur de topographies/cartes vous permet de dessiner et d'éditer des topos de manière totalement interactive. Mais n'en attendez pas trop. XTherion n'est pas un éditeur vraiment WYSIWYG. Il affiche uniquement la position et non la forme réelle des symboles de points ou de lignes dessinés. Visuellement, il n'y a pas de différence entre une excentrique et une étiquette de texte : les deux sont rendus sous forme de points simples. Le type et les autres attributs de tout objet sont spécifiés uniquement dans les menus *Contrle de point* et *Contrle de ligne*.

⁴³ L'encodage des fichiers est spécifié sur la première ligne de chaque fichier. Cette ligne est cachée par XTherion mais peut être accessible indirectement via le menu de droite.

Exercice : Trouvez deux raisons importantes pour lesquelles la topographie dessinée dans XTherion ne peut pas être identique à la sortie Therion. (Si vous répondez à cette question, vous saurez pourquoi XTherion ne sera jamais un véritable éditeur WYSIWYG. La paresse des auteurs n'est pas la bonne réponse.)

Commençons par décrire l'utilisation typique de l'éditeur de dessins. Tout d'abord, vous devez choisir la partie de la cavité à dessiner.⁴⁴



Après avoir créé un nouveau fichier dans l'éditeur de dessin, vous pouvez charger une ou plusieurs **images** (esquisses du relevé topo numérisées de la cavité⁴⁵) en tant qu'arrière-plan du dessin. Cliquez sur le bouton *Insérer* dans le menu *Images d'arrière-plan*. Malheureusement, en tant que limitation du langage Tcl/Tk, seules les images GIF, PNM et PPM (plus PNG et JPEG si vous avez installé l'extension tkImg) sont prises en charge. De plus, XTherion prend en charge le format XVI (image vectorielle XTherion), qui affiche les informations de la ligne de cheminement et les LRUD sur l'arrière-plan, ainsi que les données PocketTopo exportées au format Therion (voir ci-dessous). Toutes les images ouvertes sont placées dans le coin supérieur gauche de la zone de travail. Déplacez-les en double-cliquant et en les faisant glisser avec le bouton droit de la souris ou en utilisant le menu. Pour obtenir de meilleures performances sur des ordinateurs plus lents, il est possible de ne pas charger temporairement une image inutilisée de la mémoire en décochant la case à cocher *Visibilité*. Il est possible d'ouvrir un fichier existant sans charger d'images d'arrière-plan à l'aide du menu *Ouvrir XP*.⁴⁶

Le réglage de la taille et du zoom de la **zone de dessin** s'effectue dans le menu correspondant. Le réglage automatique (*auto adjust*) calcule la taille optimale de la zone de travail en fonction de la taille et de la position des images d'arrière-plan chargées.

Après ces étapes de préparation, vous êtes prêt pour dessiner ou, plus précisément, pour la **création d'un fichier de données cartographiques (map)**. Il est important de noter que vous créez en fait un fichier texte conforme à la syntaxe décrite dans le chapitre *Format des données*. En réalité, seul un sous-ensemble des commandes Therion est utilisé dans l'éditeur de dessins : opérations de suppression multilignes **scrap ... endscrap** pouvant contenir des commandes **point**, **ligne** et **aires**. (Cf. le chapitre *format de données*). Cela correspond à une section de la topographie dessinée à la main, constituée de points, de lignes et de zones remplies.

La première étape consiste donc à définir le **scrap** par une commande multiligne **scrap ... endscrap**. Dans le menu *Commandes de fichier*, cliquez sur le sous-menu *Action* et sélectionnez *Insérer un scrap*. Cela change le bouton *Action* en *Insert scrap* s'il avait une

⁴⁴ Il est impossible de dessiner plus d'un scrap dans un fichier, sinon dans ce cas, tous les scraps inactifs seront affichés en jaune.

⁴⁵ XTherion ne peut pas mettre à l'échelle ni imposer une rotation à des images ; En conséquences, utilisez la même orientation, échelle et résolution (DPI) pour toutes les images utilisées dans le même scrap.

⁴⁶ *Note* : Therion n'utilise pas d'images de fond tant que vous ne les assignez pas à un scrap spécifique avec l'option **-sketch**.

autre valeur. Après avoir appuyé sur ce bouton, un nouveau scrap sera insérée au début du fichier. Vous devriez voir ces nouvelles lignes dans la fenêtre preview au dessus du bouton *Insert scrap* :

```
scrap - scrap1
endscrap
end of file
```

Cette fenêtre est une description simplifiée de la structure du fichier texte tel qu'il sera enregistré par XTherion. Uniquement les commandes **scrap**, **point**, **line**, **text**—voir la raison ci-dessous— et leurs types (pour **point** et **line**) ou ID (pour **scrap**) sont affichés.

L'intégralité des commandes est affichée dans le menu *Command preview*.

Pour modifier des commandes créées précédemment, il existe des menus supplémentaires, comme par exemple, *Contrle du scrap* pour la commande du **scrap**. Ici, vous pouvez modifier l'ID (très important!) et d'autres options. Pour plus de détails, voir le chapitre *Format des données*.

Il est maintenant possible d'insérer des **symboles de points**. Comme pour l'insertion d'un scrap, accédez au menu des *commandes de fichiers*, cliquez sur le sous-menu *Action* et sélectionnez *Insérer un point*. Appuyez sur le bouton *Insérer un point* récemment renommé. Un raccourci pour tout cela est Ctrl-p. Cliquez ensuite sur l'emplacement souhaité dans la zone de travail et vous verrez un point bleu représentant un symbole de point. Ses attributs peuvent être ajustés dans le menu de *contrle de point*. Vous resterez en mode 'insertion' : chaque clic sur la zone de travail ajoute un nouveau symbole de point. Veillez à ne pas cliquer deux fois au même endroit—vous inséreriez deux symboles de point au même endroit ! Pour passer du mode 'insérer' au mode 'sélectionner', appuyez sur la touche *Echap/Esc* du clavier ou sur le bouton de *sélection* du menu *Commandes de fichiers*.

Quel sera l'ordre des commandes dans le fichier de sortie ? Exactement le même que dans le plan du menu *Commandes Fichier*. Les objets point, ligne et texte nouvellement créés sont ajoutés avant la ligne marquée dans la description de la structure. Il est possible de changer l'ordre en sélectionnant une ligne et en appuyant sur les boutons *Déplacer vers le bas*, *Déplacer vers le haut* ou *Déplacer vers* dans le menu *Commandes de fichier*. De cette façon, vous pouvez également déplacer des objets entre des scraps.

Dessiner des lignes est similaire à dessiner dans d'autres programmes de dessin vectoriel qui fonctionnent avec les courbes de Bézier. (Devinez comment entrer dans le mode d'insertion de ligne, autre que d'utiliser le raccourci Ctrl-l.) Cliquez à l'endroit où le premier point doit se situer, puis faites glisser la souris en maintenant le bouton gauche enfoncé et relâchez-le à l'emplacement où devrait se trouver le premier point de contrle. Puis cliquez ailleurs (ce point sera le deuxième point de la courbe) et faites glisser la souris (en ajustant simultanément le deuxième point de contrle de l'arc précédent et le premier point de contrle du suivant). Si cette explication semble trop obscure, on peut s'exercer à travailler avec certains programmes de dessin vectoriel standard avec une documentation

complète. La ligne sera terminée après avoir quitté le mode d'insertion. Le début et l'orientation de la ligne sont marqués par une petite coche orange à gauche au premier point.

Pour les symboles de lignes, il existe deux menus de contrôle : *Contrôle de ligne* et *Contrôle de point de ligne*. Le premier définit les attributs pour toute la courbe, comme le type ou le nom. La case à cocher *Inverser* est importante : Therion requiert des courbes orientées et il n'est pas rare que vous commenciez à dessiner du mauvais côté. Le menu de *contrôle de point de Ligne* vous permet d'ajuster les attributs de n'importe quel point sélectionné sur la ligne, par exemple une courbe lisse à cet endroit (activée par défaut) ou la présence de points de contrôle voisins (cases à cocher << et >>).

Les **Aires** sont spécifiées par leurs lignes environnantes. Cliquez sur *Insérer une zone*, puis sur les lignes entourant la zone souhaitée. Elles sont automatiquement insérées dans le champ *Aire* et nommées (s'ils ne le sont pas déjà). Une autre méthode consiste à les insérer en tant que commande **text**, dont le contenu (entré dans le menu de l'éditeur de texte de l'éditeur de carte) est la commande multilignes habituelle **area ... endarea** (voir le chapitre *Format des données*).⁴⁷

Si vous tracez des scraps sans projection, il est nécessaire de **calibrer** la zone de dessin. L'échelle ne peut être définie que d'une seule manière dans XTherion, à l'aide des coordonnées de deux points (spécifiées à la fois dans le système de coordonnées image et dans le système de coordonnées 'réel').

Après avoir sélectionné un scrap (cliquez sur son en-tête dans le menu *Commandes Fichier*), deux petits carrés rouges reliés par une flèche rouge apparaissent (par défaut, ils se trouvent dans les coins inférieurs de la zone de dessin). Vous devez les faire glisser vers des points dont les coordonnées sont connues, généralement des intersections de lignes de grille en mm sur le dessin numérisé. Si vous ne pouvez pas voir ces points, vous pouvez soit :

- Appuyer sur le bouton *Scale* dans le menu *Scraps* et cliquez à deux endroits différents de l'image où les extrémités de la flèche de calibration doivent se trouver, ou
- déplacer le pointeur de la souris sur la position souhaitée, lire les coordonnées du pointeur dans la barre d'état et entrer ces coordonnées dans les zones de *points de l'échelle de l'image* du contrôle *Scraps*. Après avoir rempli les paires de coordonnées $X1, Y1$ et $X2, Y2$, la flèche de calibrage sera déplacée en conséquence.

Ensuite, vous devez entrer les coordonnées réelles de ces points ($X1, Y1, X2, Y2$).

En **mode sélection**, vous pouvez sélectionner des objets de lignes ou de points existants et définir leurs attributs dans les menus correspondants, les déplacer ou les supprimer

⁴⁷ ATTENTION ! La commande **text** n'est pas une commande Therion, mais uniquement un hack pour un bloc de texte arbitraire dans XTherion. Dans le fichier enregistré par XTherion, il n'y aura que ce que vous entrerez dans l'*Editeur de Texte* ou ce que vous verrez dans la *Commande preview*. Cela pourra être une définition d'aire, mais aussi tout ce que vous voulez, comme par exemple un commentaire commençant avec le caractère '#'.

(Ctrl-d ou bouton *Action* dans le menu *Commandes de fichiers* après avoir défini *Action* sur *Supprimer*).

Il existe un menu *Rechercher et sélectionner* qui permet de basculer facilement d'un objet à l'autre et de visualiser ce que vous ne pouvez pas voir au premier regard sur l'image. Par exemple, si vous entrez l'expression 'station' et que vous appuyez sur *Afficher tout*, toutes les stations de l'image deviennent rouges.

XTherion ne vérifie pas la syntaxe. Il n'écrit que les objets dessinés avec leurs attributions dans un fichier texte. Toutes les erreurs sont détectées uniquement lorsque vous traitez ces fichiers avec Therion (compilation).

CONSEIL : la saisie simultanée de symboles du même type vous permet de gagner beaucoup de temps, car vous n'avez pas besoin de changer le type de symbole ni les options de remplissage pour chaque nouveau symbole. La *bote à options* conserve l'ancienne valeur saisie et il suffit donc de changer quelques caractères.⁴⁸ Il est préférable de commencer par dessiner toutes les stations topo (n'oubliez pas de leur donner des noms en fonction des noms réels dans la commande de ligne de cheminement), afin que toutes les parois de galeries soient suivies par tous les autres symboles ponctuels, lignes et zones. Enfin, dessinez des coupes transversales.

Outils additionnels

Help/Calibrate bitmap génère un fichier MaP compatible 'OziExplorer' à partir des données de géoréférencement contenues dans un fichier topographique PDF.⁴⁹ 5.3

Si la topographie au format PDF a été convertie en raster à l'aide d'un programme externe, Converter utilise une image raster *et* une topographie PDF portant le même nom de base situés dans le même répertoire pour calculer les données d'échelle.

Si le fichier PDF est utilisé directement, vous devez définir le format de DPI et de sortie avant la conversion automatique en format raster.⁵⁰

Les **données PocketTopo** exportées au format Therion⁵¹ à partir de l'application PocketTopo peuvent être importées aussi bien dans un éditeur de texte que dans un éditeur de dessin (*Fichier → Importer → Exportation Therion PocketTopo* et *Images d'arrière-plan → Insérer → Exportation Therion PocketTopo*). Le même fichier est utilisé pour les deux importations. L'importation d'une esquisse ne crée pas directement de données de scraps. Le dessin est simplement affiché sur le fond d'écran comme des bitmaps numérisés et doit être numérisé manuellement. 5.3

⁴⁸ Dans le cas des stations topographiques, XTherion incrémente automatiquement le numéro de la station à chaque nouveau symbole inséré.

⁴⁹ Les informations de calibration pour neuf points distincts sont présentes si la ligne de cheminement contient des stations fixées à l'aide d'un système de coordonnées géodésiques.

⁵⁰ **ghostscript** et **convert** doivent être installés sur votre système. Notez, que l'installation Therion pour Windows n'inclue pas **ghostscript**

⁵¹ C'est un format de texte spécial qui doit être importé à l'aide de XTherion et ne peut pas être traité directement par Therion.

Raccourcis clavier et souris dans l'éditeur de dessins

Général :

- Ctrl + Z ▷ défaire
- Ctrl + Y ▷ refaire
- F9 ▷ compiler le projet en cours
- Pour sélectionner un objet dans la liste à l'aide du clavier : utilisez la touche 'Tab' dans la liste souhaitée ; déplacez le curseur souligné sur l'objet souhaité ; appuyer sur la barre 'espace'.
- PageUp/PageDown ▷ faire défiler vers le haut / bas dans le panneau latéral.
- Shift + PageUp/PageDown ▷ faire défiler vers le haut / bas la fenêtre de commande du fichier.

Zone de dessin et images de fond :

- Clic droit ▷ défilement de zone de dessin
- Double Clic droit sur une image ▷ déplacer l'image

Insérer un scrap :

- Ctrl+R ▷ insérer un scrap

Insérer une ligne :

- Ctrl + L ▷ insérer une nouvelle ligne et entrer dans le mode 'insérer un point de ligne'
- Clic gauche ▷ insérer un point de ligne (sans points de contrôle)
- Ctrl + Clic gauche ▷ insérer un point de ligne très proche du point existant (normalement inséré juste au-dessus du point existant le plus proche)
- Clic gauche + glisser ▷ insérer un point de ligne (avec des points de contrôle)
- Ctrl enfoncée tout en faisant glisser ▷ fixer la distance du point de contrôle précédent
- Clic gauche + glisser sur le point de contrôle ▷ déplacer sa position
- Clic droit sur un des points précédents ▷ sélectionner le point précédent en mode insertion (utile si vous souhaitez également modifier la direction du point de contrôle précédent).
- Esc (Echap) ou Clic gauche sur le dernier point ▷ terminer l'insertion de ligne
- Clic gauche sur le premier point de la ligne ▷ fermer la ligne et insérer la dernière ligne

Edition d'une ligne :

- Clic gauche + glisser ▷ déplacer un point de ligne
- Ctrl + Clic gauche sur le point de contrôle + glisser ▷ déplacer le point de la ligne près du point existant (normalement, il est déplacé juste au-dessus du point existant le plus proche)
- Clic gauche sur le point de contrôle + glisser ▷ déplace le point de contrôle

Ajouter un point de ligne :

- sélectionnez le point avant lequel vous souhaitez insérer des points ; insérer les points requis; appuyez sur Echap/Esc ou faites un Clic gauche sur le point que vous avez sélectionné au début.

Effacer un point de ligne :

- sélectionnez le point que vous souhaitez supprimer ; appuyez sur *Modifier ligne* → *Supprimer un point* dans le *panneau de configuration Ligne*.

Séparation/fractionnement d'une ligne (Splitting) :

- sélectionnez le point où vous souhaitez fractionner la ligne ; appuyez sur *Modifier ligne* → *Séparer ligne* dans le *panneau de commande Ligne*.

Insertion d'un point :

- Ctrl+P ▷ permet de passer en mode 'insertion de point'.
- Clic gauche ▷ insérer un point à une position donnée.
- Ctrl + Clic gauche ▷ insérer un point très proche du point existant (normalement, il sera inséré juste au-dessus du point le plus proche).
- Esc/Echap ▷ sortir du mode 'insertion de point'.

Edition d'un point :

- LeftClick + glisser ▷ déplacer le point
- Ctrl + Clic gauche + glisser ▷ déplacer le point près du point existant (normalement, il est déplacé juste au-dessus du point existant le plus proche)
- Clic gauche + glisser les flèches de point ▷ modifier l'orientation ou la taille des points (en fonction des commutateurs donnés dans le panneau de configuration Point).

Insertion d'une aire :

- Appuyez sur les touches Ctrl + A ou les *commandes de fichier* → *Insérer* → *aire* pour passer au mode 'Insérer Aire'.
- Faites un clic droit sur les lignes qui entourent l'aire souhaitée
- Esc/Echap pour terminer l'insertion de lignes de bordure

Edition d'une aire :

- sélectionner l'aire que vous souhaitez modifier
- pressez 'Insérer' dans le 'contrle des aires' pour insérer d'autres lignes à la position actuelle du curseur
- pressez 'Insert ID' pour insérer une bordure avec un ID donné à la position actuelle du curseur
- appuyez sur 'Supprimer' pour supprimer la limite sélectionnée

Sélection d'un objet existant :

- Clic gauche ▷ sélection d'un objet en haut
- Clic droit ▷ sélectionnez un objet juste en dessous de l'objet supérieur (utile lorsque plusieurs points se superposent)

Penser Therion

Bien que tout (ou presque tout) concernant les fichiers d'entrée de Therion ait été expliqué, ce chapitre propose quelques astuces et conseils supplémentaires.

Comment entrer une ligne de cheminement (centreline) ?

La commande de base permettant la déclaration d'un bloc de lignes de cheminement est la commande `centreline`. Si la cavité est plus grande que quelques mètres, il peut être judicieux de diviser les données en plusieurs fichiers distincts et de séparer les données centreline des données de la topographie (map)

Nous utilisons classiquement un seul fichier `*.th` qui contient un bloc `centreline` par séance topographique. Il est facile de démarrer avec un fichier template comme celui montré ci-dessous, où les points seront remplacés par le texte approprié.

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

Pour créer un espace de nom unique, la commande `centreline` est incluse dans la commande `survey ... endsurvey`. C'est utile lorsque le relevé porte le même nom que le fichier qui le contient.⁵² Les points seront ensuite référencés avec le caractère `@`—voir la description de la commande `survey`.

Pour les très grandes cavités, il est possible de construire une structure hiérarchique des répertoires. Dans un tel cas, nous créons un fichier spécial appelé `INDEX.th`, qui contient tous les autres fichiers `*.th` d'un répertoire donné et contient aussi des commandes `equate` permettant de définir les connexions entre les topographies.

Comment dessiner des topographies ?

Le plus important est de concevoir la division de la cavité en plusieurs scraps. Le `Scrap` est la pierre angulaire de la topographie. C'est généralement une *mauvaise* idée d'essayer d'ajuster chaque scrap au fichier `*.th` correspondant. La raison en est que les connexions entre les scraps doivent être aussi simples que possible. En règle générale, les scraps sont indépendants de la hiérarchie de la ligne de cheminement. Essayez donc d'oublier la hiérarchie des relevés lorsque vous tracez des topographies et choisissez les meilleures jonctions.

Nous insérons généralement des topographies dans l'avant-dernier niveau de la hiérarchie du relevé.⁵³ Chaque scrap peut contenir une partie arbitraire de n'importe quelle topographie du dernier niveau hiérarchique. Par exemple, une topographie principale contient les topographies `a`, `b`, `c` et `d`. Les relevés `a` – `d` contiennent les données de la ligne de cheminement de quatre secteurs du relevé et chacun d'eux se trouve dans un fichier séparé. Il existe une topographie `main_map` qui contient les scraps `s1` et `s2`. Si la topo principale (`main_map`) est située dans le relevé principal (`main`), le scrap `s1` peut couvrir une partie de la ligne de cheminement de la topographie `a`, compléter le relevé `b` et une partie de `c` ; `s2` couvrira une partie des topographies `a` et `c` ainsi qu'une topographie complète `d`. Les noms des stations du relevé seront référencés à l'aide du symbole `@` (par exemple `1@a`) dans les scraps.⁵⁴

Les scraps sont généralement stockés dans des fichiers `*.th2`. Chaque fichier peut contenir plusieurs scraps. Pour garder les données bien organisées, il existe quelques conventions de dénomination : dans le fichier `foo.th2`, tous les scraps sont nommés `foo_si`, où `i` est égal à `1`, `2`, etc. Les sections transverses sont nommées `foo_ci`, les lignes `foo_li`, etc. Cela aide beaucoup avec les grands systèmes karstiques : si un scrap est référencé, vous savez immédiatement dans quel fichier il a été défini.

⁵² E.g. `survey entrance` dans le fichier `entrance.th`.

⁵³ N'oubliez pas que les relevés créent des espaces de noms. Vous ne pouvez donc référencer que les objets du relevé et de tous les sous-niveaux.

⁵⁴ Si vous incluez des topographies dans le relevé de niveau supérieur, vous pouvez référencer n'importe quelle station dans n'importe quelle feuille, ce qui est très flexible. Mais en revanche, vous devrez utiliser des noms plus longs dans les références de stations, comme `3@jno.katakombi.jmn.dumbier`

Comme pour les fichiers `*.th`, il peut y avoir un fichier `INDEX.th2` par répertoire qui inclut tous les fichiers `*.th2`, définit les jointures de scraps et les topographiess finales.

Lorsque vous tracez des scraps, vérifiez si le contour est correctement défini : toutes les lignes créant la bordure extérieure doivent avoir l'option `-outline out` ; toutes les lignes entourant les piliers intérieurs l'option `-outline in`. Les contours du scrap ne peuvent pas se croiser, sinon la face intérieure du scrap ne peut pas être déterminée. Il y a deux tests simples pour savoir si le contour du scrap est correct :

- il n'ya aucun message d'avertissement (warning) METAPOST “`scrap outline intersects itself`” (le contour du scrap se croise)
- lorsque vous définissez le remplissage des passages/galeriess sur n'importe quelle couleur (option `color map-fg <nombre>` dans le `layout`), vous pouvez voir ce que Therion considère comme se trouvant à l'intérieur du scrap.

Comment construire les modèles ?

Un modèle est créé à partir des contours des scraps. La hauteur et la profondeur des galeries sont calculées à partir des symboles de points `passage-height` et `dimensions` (dans les scraps).

Therion en profondeur

Comment le dessin est-il construit ?

Ce chapitre explique comment les options `-clip`, `-place`, `-visibility` et `-context` des commandes `point`, `line` et `area` fonctionnent exactement. Il explique également les options `color`, `transparency`, `symbol-hide` et `symbol-show` (respectivement couleur, transparence, masque et affichage de symboles) de la commande de présentation `layout`.

Lors de l'exportation de la topographie, Therion doit déterminer trois attributs pour chaque symbole de point, de ligne ou de zone : visibilité (`visibility`), coupure (`clip`) et ordre (`place` ; `orderring`).

(1) Le symbole est visible si tout ce qui suit est vrai :

- l'option `-visibility on` est activée (par défaut pour tous les symboles)
- il n'a pas été masqué par l'option `-symbol-hide` dans la présentation (`layout`),
- si son option `-context` est définie, le symbole correspondant n'a pas été masqué par l'option `-symbol-hide` de la présentation (`layout`).

Seuls les symboles visibles sont exportés.

(2) Certains symboles sont coupés par le contour du scrap. Ce sont par défaut tous les éléments suivants :

- *symboles points* : symboles de remplissages de galeries/passages (substrat rocheux... guano),
- *symboles lignes* : tous les symboles de ligne qui n'ont pas l'option `-outline` mentionnée, à l'exception de `section`, `arrow`, `label`, `gradient` et `water-flow`
- *symboles d'aires* : tous.

Le paramètre par défaut peut être modifié à l'aide de l'option `-clip`, si cela est autorisé pour un symbole particulier. Tous les autres symboles ne sont pas coupés par la limite du scrap.

(3) Ordre : Chaque symbole appartient à l'un des groupes suivants, qui sont dessinés les uns après les autres :

- bottom (bas) ▷ tous les symboles avec l'option `-place bottom`
- default-bottom ▷ tous les symboles area par défaut
- default ▷ symboles qui n'appartiennent à aucun autre groupe
- default-top ▷ `ceiling-step` (marche de plafond) et `chimney` (cheminée) par défaut
- top ▷ tous les symboles avec l'option `-place top`

L'ordre des symboles à l'intérieur de chaque groupe suit l'ordre des commandes dans le fichier d'entrée⁵⁵ : les symboles qui viennent en premier sont dessinés en dernier (c'est-à-dire qu'ils sont affichés en haut de chaque groupe).

Nous sommes maintenant prêts à décrire comment les topographies (ou les chapitres de l'atlas) sont construites :

- chaque aire de la topographe finale est remplie grâce à `color map-bg`
- les surfaces bitmap sont affichées si `surface` est configurée avec `bottom`
- FOR (pour) chaque scrap : le contour est rempli de blanc
- la grille est affichée si `grid` est configurée avec `bottom`
- preview below (aperçu ci-dessous)⁵⁶ est rempli grâce à `color preview-below`
- FOR (pour) chaque niveau⁵⁷ :

BEGIN (début) de la transparence

FOR (pour) chaque scrap : le contour est rempli grâce à `color map-fg`

FOR (pour) chaque scrap : les symboles d'aires sont remplis et soudés (clip) à la limite du scrap

⁵⁵ Ou menu *File commands* dans XTherion

⁵⁶ comme spécifié en utilisant l'option `preview` dans la commande `map`.

⁵⁷ Level est une collection de scraps non séparés par une commande `break` dans la commande `map`

END (fin) de la transparence

BEGIN (début) de la soudure des étiquettes de texte (pour toutes les étiquettes dans ce niveau et les niveaux supérieurs)

FOR chaque scrap :

dessiner tous les symboles à découper (à l'exception de `line survey`)

mettre en ordre de bas en haut

tracer les symboles de `line survey` qui sont coupés en bordure de scrap

découper à la bordure du scrap

FOR chaque scrap:

dessiner tous les symboles à ne pas couper (à l'exception des points de stations `point station` et de toutes les étiquettes)

mettre en ordre de bas en haut

dessiner les symboles `point station`

END (fin) du découpage par les étiquettes de texte

FOR (pour) chaque scrap : dessiner toutes les étiquettes (point et ligne) (y compris `wall-altitude`)

- l'aperçu ci-dessus est dessiné avec `color preview-above`
- les bitmaps de surface sont affichés si `surface` est positionné sur `top`
- la grille est affichée si `grid` est positionné sur `top`

Nous marchons tous les deux et ne marchons pas dans les mmes rivires.
Ποταμοῖς τοῖς αὐτοῖς ἐμβαίνομέν τε καὶ οὐκ ἐμβαίνομεν.
—Heraclitus of Ephesus, 6^{me}/5^{me} sicle av-JC

Traitement des donnes

Outre les fichiers de donnes contenant des donnes du relev, Therion utilise un fichier de configuration contenant des instructions sur la prsentation des donnes.

Fichier de configuration

Le nom de fichier de configuration peut tre donn comme argument Therion. Par dfaut, Therion recherche le fichier nomm `thconfig` dans le rpertoire de travail en cours. Il se lit comme tout autre fichier Therion (cest--dire une commande par ligne ; les lignes vides ou commençant par ‘#’ sont ignores ; les lignes terminées par une barre oblique inverse (backslash) se poursuivent la ligne suivante). Une liste des commandes actuellement prises en charge est donnée ci-après :

‘system’

Permet d’exécuter des commandes système lors de la compilation de Therion.⁵⁸ Normalement, Therion attend que le sous-processus soit termin. Si vous souhaitez continuer la compilation sans interruption, utilisez l’expression `<command> &` sous Linux et l’expression `<command>` sous Windows.

‘encoding’

`encoding` fonctionne comme la commande d’encodage dans les fichiers de donnes—il spécifie les jeux de caractères.

‘language’

Syntaxe :

- `language <xx_[YY]>`

Définit la langue de sortie pour les textes traduisibles.

5.3

⁵⁸ Par exemple pour ouvrir ou refresh un lecteur PDF externe.

'cs'

Syntaxe :

- `cs <systeme de coordonnees>`

5.3 En dehors du bloc `layout`, la commande spécifie le système de coordonnées de l'export. Il n'est pas possible de spécifier différents systèmes de coordonnées pour différents exports (La dernière occurrence de `cs` est utilisée pour tous les exports).

Si `cs` n'est pas définie dans le fichier de configuration, alors, la première commande `cs` trouvée par Therion lors de la lecture des données sera utilisée pour les exports.

A l'intérieur d'un `layout`, cette option spécifie le système de coordonnées pour la localisation des données associées (`origin`, `grid-origin`).

'sketch-warp'

Syntaxe :

- `sketch-warp <algorithm>`

Spécifie quel algorithme de morphing utiliser pour déformer les scraps. Plusieurs algorithmes sont possibles : `line`—par défaut ; `plaquette`—inventé par Marco Corvi.

'input'

Fonctionne comme la commande `input` dans les fichiers de données—Inclus d'autres fichiers

'source'

Description : Spécifie les fichiers sources (données) que Therion doit lire. Vous pouvez spécifier plusieurs fichiers ici ; un par ligne. Vous pouvez également les spécifier à l'aide de l'option de ligne de commande `-s` (voir ci-dessous).

Il est également possible de taper (quelques petits extraits de code) directement dans le fichier de configuration en utilisant la syntaxe multiligne.

Syntaxe :

```
source <nom_de_fichier>
```

or

```
source
```

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

```
endsource
```

Arguments :

- `<nom_de_fichier>`

‘select’

Description : Slectionne les objets (relevs et dessins) exporter. Par dfaut, tous les objets du relev sont slectionnns. Si aucune topographie n’est slectionne, toutes les notes provenant des topographies slectionnees sont slectionnees par dfaut pour l’exportation de la topographie.

S’il n’y a pas de scraps ou de topos dans les donnees, la ligne de cheminement de toutes les topographies est exporte dans la topographie.⁵⁹

Lorsque vous exportez des topographies dans diffrentes projections, vous devez les slectionner sparment pour chaque projection

Syntaxe : `select <object> [OPTIONS]`

Arguments :

- `<object>` ▷ tout relev ou topographie identifis par leur identifiant ID.

Options :

- `recursive <on/off>` ▷ valide uniquement lorsqu’une topographie est slectionne. Si cette option est active (par dfaut), tous les sous-niveaux du relev donn sont slectionnns / dslectionnns de manire rcursive.
- `map-level <number>` ▷ valide uniquement lorsqu’une topo est slectionne. Dtermine le niveau auquel l’extension de la topographie pour l’exportation d’atlas est arrte. Par dfaut, 0 est utilis. si ‘basic’ est spcifi, l’extension se fait jusqu’aux topographies de base.
Remarque : les aperus de ne sont affichs que comme spcifi dans les topographies du `map-level` en cours.
- `chapter-level <number>` ▷ valide uniquement lorsqu’une topographie est slectionne. Dtermine le niveau (level) auquel, lors de l’exportation de l’atlas, l’extension de chapitre sera stoppe. Par dfaut, 0 est utilis. Si vous spcifiez ‘-’ ou ‘.’, aucun chapitre nest export pour cette topographie. Si l’option de pages de titre (`title-pages`) de la mise en page (`layout`) est active, chaque chapitre commence par une page de titre.

‘unselect’

Description : Dslectionne les objets exporter.

Syntaxe : `unselect <object> [OPTIONS]`

Arguments :

Identique la commande `select`.

⁵⁹ ATTENTION : s’il y a normment de donnees, ce qui est le cas, avec des cavits ou des systmes fort dveloppement, la mmoire est rapidement sature et le compilation crashe.

Options :

Identique la commande `select`.

‘text’

Description : Spécifie la traduction de tout texte Therion par défaut dans le fichier de sortie.

Syntaxe : `text <language ID> <texte therion> <mon texte>`

Arguments :

- `<language ID>` ▷ identifiant de langue ISO standard (par exemple `fr`, `en` ou `en_GB`)
- `<therion text>` ▷ texte Therion traduire. Pour une liste des textes Therion et des traductions disponibles, voir le fichier `thlang/texts.txt`.

‘layout’

Description : Spécifie les caractéristiques de mise en page de l’export pour les topographies 2D. Les settings qui s’appliquent au mode atlas sont marqués ‘A’ ; au mode map ‘M’.

Syntaxe : `layout <id> [OPTIONS]`

```
copy <source layout id>
cs <système de coordonnées>
north <true/grid>
scale <longueur sur l'image> <longueur réelle>
base-scale <longueur sur l'image> <longueur réelle>
units <metric/imperial>
rotate <nombre>
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> <normalsize> <largesize> <hugesize>
size <width> <height> <units>
overlap <valeur> <unites>
page-setup <dimensions> <unites>
page-numbers <on/off>
exclude-pages <on/off> <liste_des_pages>
title-pages <on/off>
nav-factor <facteur>
```

```

nav-size <x-size> <y-size>
transparency <on/off>
opacity <valeur>
surface <top/bottom/off>
surface-opacity <valeur>
sketches <on/off>
layers <on/off>
grid <off/top/bottom>
grid-origin <x> <y> <x> <unites>
grid-size <x> <y> <z> <unites>
grid-coords <off/border/all>
origin <x> <y> <z> <unites>
origin-label <x-label> <y-label>
own-pages <nombre>
page-grid <on/off>
legend <on/off/all>
legend-columns <nombre>
legend-width <n> <unites>
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> <nom_de_fichier>
statistics <explo/topo/carto/copyright all/off/nombre>
            <explo/topo-length on/off>
scale-bar <longueur> <unites>
survey-level <N/all>
language <xx[_YY]>
colour/color <item> <couleur>
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>
endcode
endlayout

```

Arguments :

<id> ▸ identifiant du layout (utilis par la commande **export**)

Command-like options :

- **copy <source layout id>** ▸ dfinissez ici les propri s qui ne sont pas modifi s en fonction de la pr sentation source donne **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)

5.4 CCNP (Carlsbad Caverns National Park)

5.4 NZSS (New Zealand Symbol Set)

SKBB (Speleoklub Banská 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)

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

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.

5.4 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.

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)
- `symbol-colour <point/line/area/group/special> <symbol-type> <colour>` ▸ 5.3
change colour of particular symbol or group of symbols.⁶⁰ (M, A)
- `min-symbol-scale <scale>` ▸ define minimal `<scale>`, from which points and lines 5.4.1
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.
- `fonts-setup <tinysize> <smallsize> <normalsize> <largesize> <hugesize>` ▸ 5.4.1
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
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`)

⁶⁰ Note: colour change currently applies to pattern fills only if (1) output format is PDF and (2) META-POST version is at least 1.000

- **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 (astromonomical) 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 image⁶¹ 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

⁶¹ Note that you can include PDF too, which may be used to combine plan and extended elevation into one nice looking PDF file.

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’

Syntaxe :

- `setup3d <value>`

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

‘sketch-colors’

Syntaxe :

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

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

‘export’

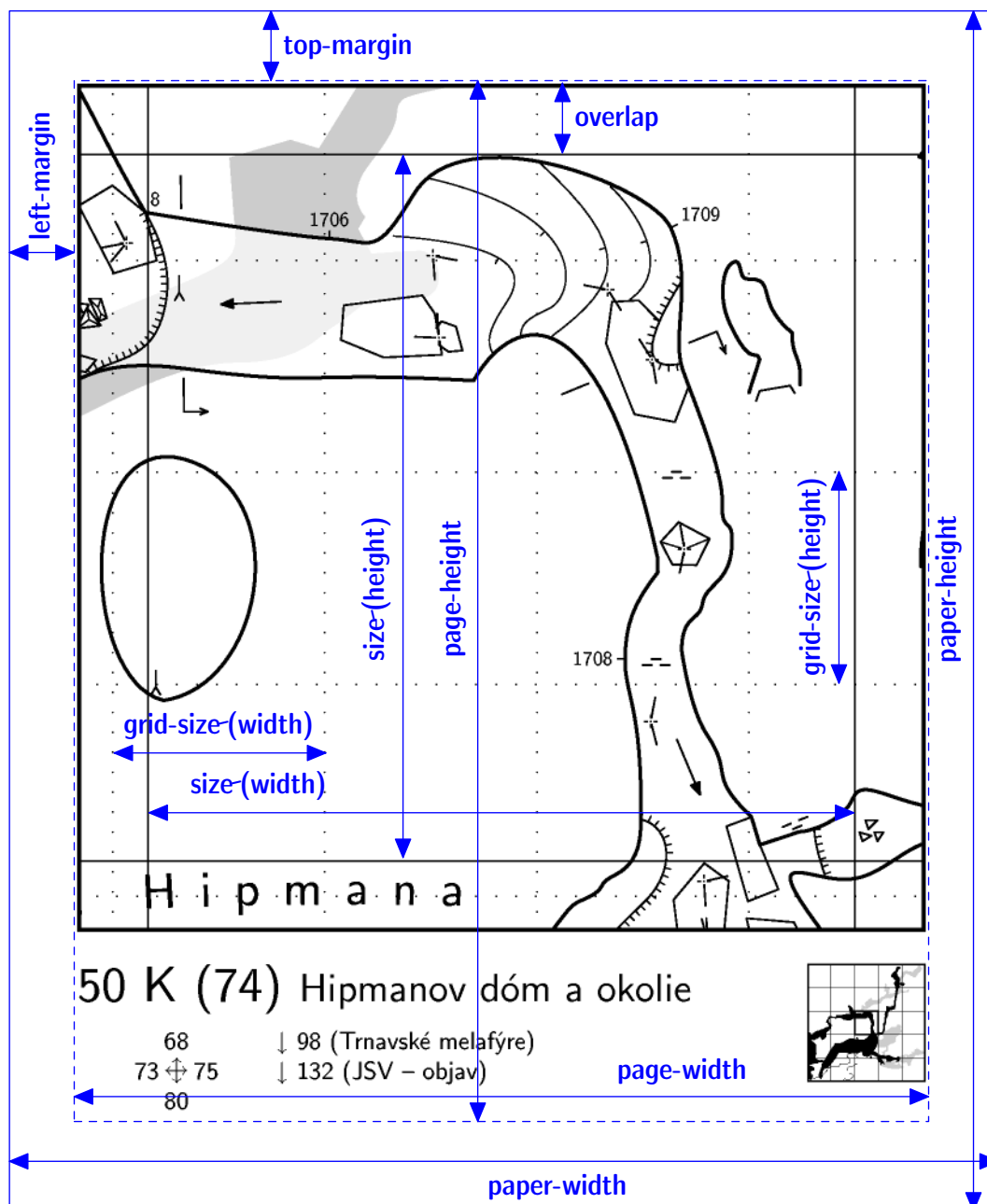
Description : Exports selected surveys or maps.

Syntaxe :

- `export <type> [OPTIONS]`

Arguments :

- `<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

Options :

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), `dxg`, `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`⁶², `survex`, `dxg`, `esri`⁶³, `kml` (Google Earth), `xvi`⁶⁴ and `bbox`⁶⁵ 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.

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

⁶² SVG embedded in XHTML file which contains also legend

⁶³ ESRI shapefiles. Multiple files are written to a directory with the specified filename.

⁶⁴ 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.

⁶⁵ Text file containing geographic coordinates of lower-left and upper-right corners of the map area.

common for lists:

- 5.4 • **format/fmt** <format> ▷ set continuation output format. Currently the following output formats are supported: **html** (default), **txt**, **kml**⁶⁶ 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. 5.3

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. 5.3

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	
map	pdf, svg, xhtml, dxf, esri, survex, xvi, kml, bbox	5.3
atlas	pdf	
database	sql, csv	5.4
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

therion

The full syntax is

```
therion [-q] [-L] [-l <log-file>]
        [-s <source-file>] [-p <search-path>]
        [-b/--bezier]
        [-d] [-x] [--use-extern-lib] [<cfg-file>]
```

or

⁶⁶ For cave-list and continuation-list.

```

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

```

Arguments :

`<cfg-file>` Therion takes only one optional argument: the name of a configuration file. If no name is specified `thconfig` in the current directory is used. If there is no `thconfig` file (e.g. current directory is not a data directory), Therion exits with an error message.

Options :

- `-d` ▷ Turn on debugging mode. The current implementation creates a temporary directory named `thTMPDIR` (in your system temporary directory) and does not delete any temporary files.
- `-h, --help` ▷ Display short help.
- `-L` ▷ Do not create a log-file. Normally therion writes all the messages into a `therion.log` file in the current directory.
- `-l <log-file>` ▷ Change the name of the log file.
- `-p <search-path>` ▷ This option is used to set the search path (or list of colon-separated paths) which therion uses to find its source files (if it doesn't find them in the working directory).
- `-q` ▷ Run therion in quiet mode. It will print only warning and error messages to STDERR.
- `--print-encodings` ▷ Print a list of all supported input encodings.
- `--print-tex-encodings` ▷ Print a list of all supported encodings for PDF output.
- `--print-init-file` ▷ Print a default initialization file. For more details see the *Initialization* section in the *Appendix*.
- `--print-environment` ▷ Print environment settings for therion.
- `--print-symbols` ▷ Print a list of all therion supported map symbols in `symbols.xhtml` file.
- `-s <source-file>` ▷ Set the name of the source file.
- `--use-extern-libs` ▷ Don't copy T_EX and METAPOST macros to working directory. T_EX and METAPOST should search for them on their own. Use with caution.

- `-v, --version` ▷ Display version information.
- `-x` ▷ Generate file ‘.xtherion.dat’ with additional information for XTherion.

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 $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$, where Δx 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.

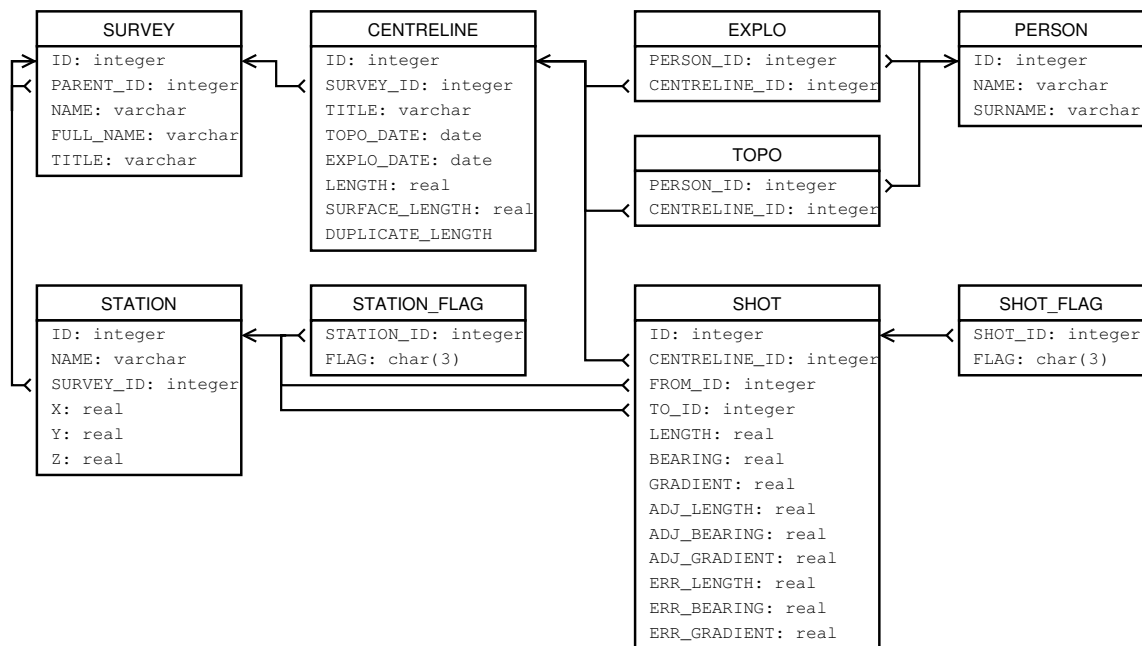


Table and column names are self-explaining; for undefined or non-existing values `NULL` is used. `FLAG` in `SHOT_FLAG` table is `dpl` or `srf` for duplicated or surface shots; in `STATION_FLAG` table `ent`, `con`, `fix`, `spr`, `sin`, `dol`, `dig`, `air`, `ove`, `arc` for stations with entrance, continuation, 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 marked⁶⁷ 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.

⁶⁷ using `station` attribute for centreline point and `point continuation` in scraps

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.

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

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). 5.3

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 Survex 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

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

The T_EX 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` ▷ T_EX 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`, `\ss`, `\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\pointerW\hskip0.7em}%
              \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 T_EX macros or T_EX 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
- 5.4 • `\cavemaxz`, `\caveminz` ▷ altitude max/min value
- 5.4 • `\thversion` ▷ current therion version
- 5.4 • `\currentdate` ▷ current date
- 5.4 • `\outcodes`, `\outcsname` ▷ 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

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 two⁶⁸ 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

There is a general-purpose macro for typesetting in multiple columns⁶⁹:

- `\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:

⁶⁸ Default; adjust the `legend-columns` layout option to get them more or less

⁶⁹ Not to be used with map legend, where multiple columns are to be adjusted by `legend-columns` layout option

```

\cavename, \comment, \copyrights, \exploitle, \topotitle, \cartotitle,
\exploteam, \topoteam, \cartoteam, \exploitle, \topodate, \cartodate,
\cavelength, \cavedepth, \cavelengthtitle, \cavedepthtitle, \cavemaxz,
\caveminz, \thversion, \currentdate, \outcode, \outcode, \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:

```

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` dimen 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/cstr/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 ¹1986

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' subtype 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`.

The following variables are also available:

- boolean `ATTR__shotflag_splay`, `ATTR__shotflag_duplicate`,
`ATTR__shotflag_approx` ▷ set for line survey

5.4

- 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 METAPOST (<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.⁷⁰ Copy an appropriate encoding mapping file with an `*.enc` extension to the `texenc/encodings`, run the Perl script `mktextenc.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. Predefined macros are compiled into the Therion executable and are copied to the working directory just before running METAPOST and T_EX (unless the `--use-extern-lib` 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.

⁷⁰ 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 pdfTEX 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 TEX. Useful if there is other TEX distribution installed on your system which had set-up any environment variables, which could confuse TEX and METAPOST programs supplied in Therion for Windows distribution.

Set to `on` if you use other TEX 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>`

5.4

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 pdfTeX 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 (pdfTeX 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 (pdfTeX 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)
pdfTeX 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
endline

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>)