

## ctools - Feature #1816

### Should we store parameter error maps when running cttsmap?

07/18/2016 01:17 PM - Mayer Michael

<b>Status:</b>	Closed	<b>Start date:</b>	07/18/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Mayer Michael	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.5.0		
<b>Description</b>			
<p>At the moment, when running cttsmap, the tool creates one additional map for each free source parameter in the fit. In order to provide more detailed information, we could add maps for each parameter error as well (it could be steered via a hidden user parameter). Furthermore, one might want to store the fit status per bin to check if there were fits that did not converge.</p> <p>Additionally, it might make sense to also be able to compute upper limit maps. This however could be a separate tool, e.g. ctulimmap?</p>			

#### History

##### #1 - 07/18/2016 10:41 PM - Knödseder Jürgen

Fully support to write out also the error maps in cttsmap.

##### #2 - 07/18/2016 10:42 PM - Knödseder Jürgen

user#77 wrote:

Additionally, it might make sense to also be able to compute upper limit maps. This however could be a separate tool, e.g. ctulimmap?

Agree that this should be a separate tool, maybe ctulmap or ctulimitmap?

##### #3 - 07/19/2016 09:51 AM - Mayer Michael

- Assigned To set to Mayer Michael

Ok, I will implement the error maps in cttsmap and come up with a first version of ctulimitmap.  
What is your feeling about the fit status as an additional map?

##### #4 - 07/29/2016 09:42 AM - Mayer Michael

I propose to use the prefix "e\_" in the FITS extension name in order to indicate that we deal with an error map. The maps in the TS map would thus be e.g. "PhotonFlux", "e\_PhotonFlux", ...  
Does this sound reasonable?

##### #5 - 07/29/2016 10:05 AM - Knödseder Jürgen

user#77 wrote:

I propose to use the prefix "e\_" in the FITS extension name in order to indicate that we deal with an error map. The maps in the TS map would thus be e.g. "PhotonFlux", "e\_PhotonFlux", ...  
Does this sound reasonable?

I think I agree. I looked some time ago in how catalogues in Vizier use uncertainty information on quantities, and the uncertainty columns are generally prepended by `e_`. It thus makes also sense to me to extent this convention to the extension names.

We may write down somewhere the convention, probably in a data format specification for GammaLib and ctools that does not yet exist.

And we may also make sure that this convention is always respected (cspull does not - it uses `Unc_`, `csspec` and `cslightcrv` seem to respect it). We probably want to create a change request for that.

It may also be useful to discuss this with the data format group.

**#6 - 08/01/2016 02:10 PM - Mayer Michael**

- *Status changed from New to Pull request*

Ok, I have implemented the changes adding the errors using the prefix `"e_"`. I agree this should be discussed further. In particular, we might want to deal with asymmetric errors eventually. My proposal is `"e_"` for symmetric errors and `"ed_"` and `"eu_"` for asymmetric ones but I have no strict opinion on that.

The changes include:

- `cttsmap` now has a hidden boolean parameter `errors`, which signals that errors on the free parameters should be computed and stored (default=no, since it will cost quite some additional computation time).
- If `errors=yes` for each parameter and additional error map with the prefix `"e_"` in the extension name is appended.
- I added a map called `"FITSTATUS"` that shows the fit status of each optimisation on the grid (provided by `opt->status()`).
- I renamed `"STATUS MAP"` in `"PROGRESS MAP"` in order not to be confused with the `FITSTATUS` map
- Adapted `cstsmmerge` accordingly.
- Added unit test for `cttsmap` using `errors=yes`.

The branch is:

`1816-cttsmap-errors`

(I will create a separate issue for `ctlimitmap`).

**#7 - 08/04/2016 10:28 AM - Mayer Michael**

- *% Done changed from 0 to 100*

**#8 - 08/05/2016 12:18 AM - Knödseder Jürgen**

user#77 wrote:

Ok, I have implemented the changes adding the errors using the prefix `"e_"`. I agree this should be discussed further. In particular, we might want

to deal with asymmetric errors eventually. My proposal is "e\_" for symmetric errors and "ed\_" and "eu\_" for asymmetric ones but I have no strict opinion on that.

The changes include:

- cttsmmap now has a hidden boolean parameter errors, which signals that errors on the free parameters should be computed and stored (default=no, since it will cost quite some additional computation time).
- If errors=yes for each parameter and additional error map with the prefix "e\_" in the extension name is appended.
- I added a map called "FITSTATUS" that shows the fit status of each optimisation on the grid (provided by opt->status()).
- I renamed "STATUS MAP" in "PROGRESS MAP" in order not to be confused with the FITSTATUS map
- Adapted cttsmmapmerge accordingly.
- Added unit test for cttsmmap using errors=yes.

The branch is:

*1816-cttsmmap-errors*

(I will create a separate issue for ctulimitmap).

Thanks. I had not realised that the errors were not available in the TS computation. Good idea to make the error computation optionally.

I was wondering whether the "FITSTATUS" and the "PROGRESS MAP" maps are not redundant? Both are set at the exit of the fit, one with the fit status, the other with 1. Possible fit status values are:

```
#define G_LM_CONVERGED      0
#define G_LM_STALLED       1
#define G_LM_SINGULAR       2
#define G_LM_NOT_POSTIVE_DEFINITE 3
#define G_LM_BAD_ERRORS    4
```

You could either add some positive value (e.g. 1 or 10) to get non zero values. Or you may default the map to -1 for non-fitted values. And you could keep the original "STATUS MAP".

**#9 - 08/05/2016 02:05 PM - Mayer Michael**

I agree that the maps are in fact redundant. I haven't thought about this before.

You could either add some positive value (e.g. 1 or 10) to get non zero values. Or you may default the map to -1 for non-fitted values. And you could keep the original "STATUS MAP".

I went for the latter and initialised the "STATUS MAP" with -1 in `cttsmap::init_maps()`. This map is now used in `cstsmmerge` to check for the computation progress.

I rebased branch `1816-cttsmap-errors` on `devel`, and started to adapt `cstsmmerge` to the new logging convention. I modified and extend the unit tests a bit and added appropriate example test files.

I guess we could merge this in now.

#### #10 - 08/05/2016 02:28 PM - Mayer Michael

P.S. I forgot to mention that for some reason I wasn't able to track down we now have blank lines in the log files after using:

```
self._log_value(gammalib.TERSE, "Parameter", value)
```

`GApplication` only seems to do `std::endl` but not insert an additional blank line. Any ideas?

#### #11 - 08/05/2016 04:52 PM - Knödlseher Jürgen

Is also fixed already. Sorry for that. The new methods induced some reorganisation of things.

Checkout the latest code and see if it's gone.

Btw: `log_string()` now automatically appends a linefeed since it turned out that in basically every line I had to add one by hand. You can still switch this off by using

```
log_string(NORMAL, "Bla", false);
```

I also try to fix the same logic for the chattiness levels. Is not yet done for the `cscripts`. The rule is:

- TERSE: only for headers (running a tool just shows the steps, and the output in `ctulimit` as this is not written into a file, hence otherwise useless)
- NORMAL: normal behavior, only summary information and results
- EXPLICIT: more detailed information, in particular about computing things, but also content of for example an observation container; use `m_obs.print(m_chatter)` to forward the chattiness to `GammaLib` classes
- VERBOSE: whatever might be useful for diagnostics, mainly to debug things, or to look into the detailed inner workings; rarely used

I have to write this up in the coding conventions.

**#12 - 09/26/2016 12:13 PM - Knödlseider Jürgen**

Michael, can you confirm that this is fixed now (want to close the issue)?

**#13 - 09/26/2016 12:58 PM - Mayer Michael**

It seems that the above branch (*1816-cttmap-errors*) is not yet merged into the devel branch. At least there is not errors parameter in the parameter file yet.

**#14 - 09/26/2016 10:55 PM - Knödlseider Jürgen**

- *Status changed from Pull request to Feedback*

Merged into devel. There were some merge conflicts since I also modified the code on my side. You may check if everything works fine. I will put the status for the moment to Feedback.

**#15 - 09/27/2016 02:47 PM - Mayer Michael**

- *File tsmmap.py added*

- *File tsmmap.fits added*

I have made some checks and it seems to work nicely. However, I discovered some disturbing results when creating a simple TS map from a simulated CTA observation on the Crab Nebula.

All TS values of bins that are not close to the source are largely negative (range -1000 to -600). I guess this is not intended. I attached the script to reproduce this result. Not sure this is a feature of cttmap or something more profound.

**#16 - 09/27/2016 03:19 PM - Knödlseider Jürgen**

user#77 wrote:

I have made some checks and it seems to work nicely. However, I discovered some disturbing results when creating a simple TS map from a simulated CTA observation on the Crab Nebula.

All TS values of bins that are not close to the source are largely negative (range -1000 to -600). I guess this is not intended. I attached the script to reproduce this result. Not sure this is a feature of cttmap or something more profound.

I guess it's more a feature that comes from the fact that the Crab flux never can go to zero. I see that the minimum prefactor is  $1e-23$ . I guess if you set this to  $1e-30$  the negative TS become much less negative (though never 0). Note also that the index goes to -5 when you move array from the source.

**#17 - 09/27/2016 05:09 PM - Mayer Michael**

I just tested it down to using  $1e-40$  as minimum prefactor. I also set the starting value of the prefactor to  $1e-23$ . Both approaches revealed TS values in the -1000 range.

**#18 - 03/03/2017 10:35 AM - Knödlseider Jürgen**

- *Target version changed from 1.2.0 to 1.3.0*

#19 - 06/07/2017 05:46 PM - Knödlseider Jürgen

- Target version changed from 1.3.0 to 1.4.0

#20 - 08/01/2017 09:51 AM - Knödlseider Jürgen

- Status changed from Feedback to In Progress

- Target version changed from 1.4.0 to 1.5.0

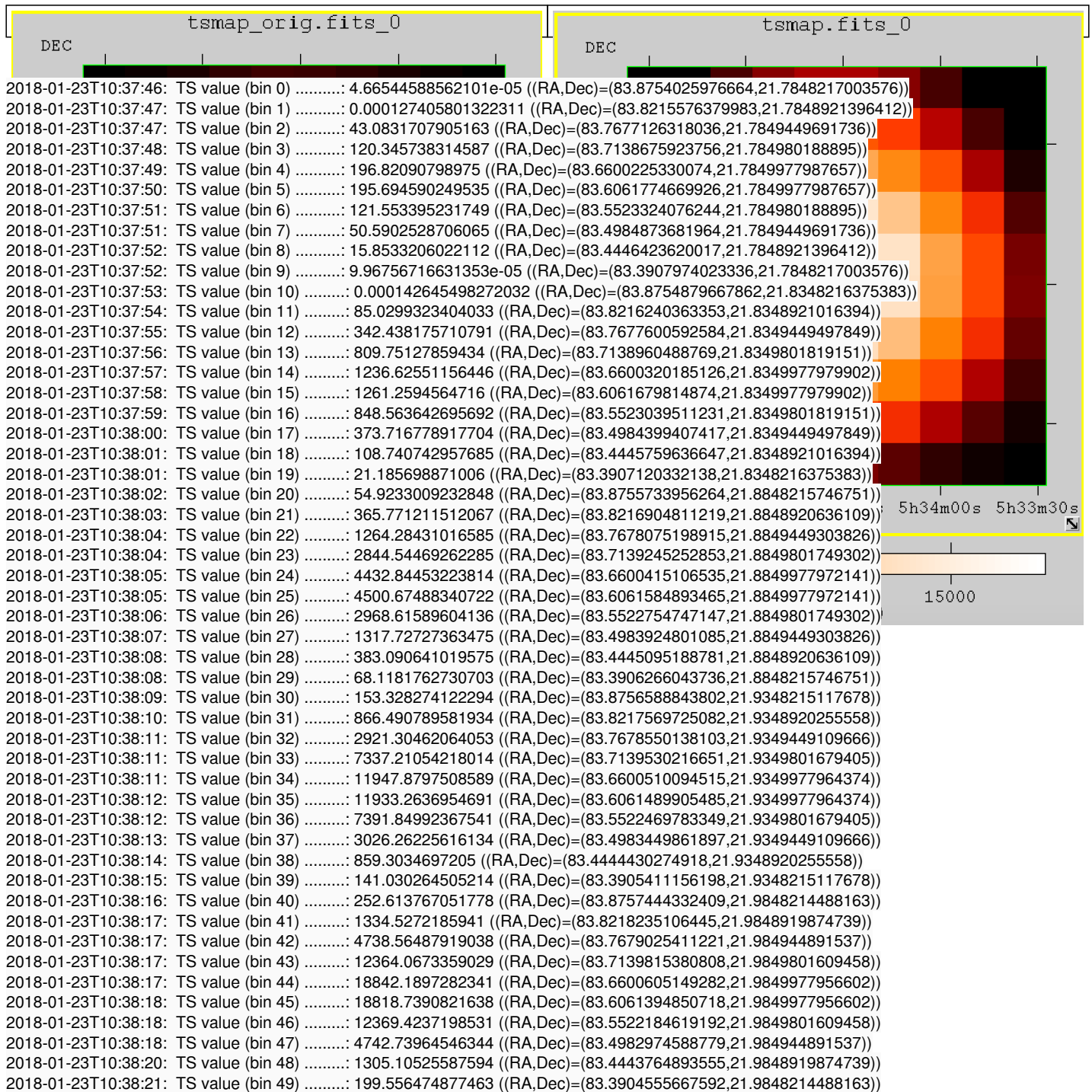
- % Done changed from 100 to 90

#21 - 01/23/2018 11:44 AM - Knödlseider Jürgen

- File tsmap.png added

- File tsmap\_orig.png added

In the actual version of the software I could not reproduce the issue. On the left the original TS map, on the right the current TS map. Although it cannot be seen directly, the current TS map has no negative values (below also the values from the log file).



2018-01-23T10:38:21: TS value (bin 50) .....: 264.137835760921 ((RA,Dec)=(83.8758300424022,22.0348213858204))  
 2018-01-23T10:38:22: TS value (bin 51) .....: 1391.64779917573 ((RA,Dec)=(83.8218900956817,22.0348919493651))  
 2018-01-23T10:38:23: TS value (bin 52) .....: 4968.01272047381 ((RA,Dec)=(83.7679501019346,22.0349448720938))  
 2018-01-23T10:38:23: TS value (bin 53) .....: 12817.1487023782 ((RA,Dec)=(83.714010074597,22.0349801539462))  
 2018-01-23T10:38:23: TS value (bin 54) .....: 19358.0839990357 ((RA,Dec)=(83.6600700271051,22.0349977948825))  
 2018-01-23T10:38:23: TS value (bin 55) .....: 19339.7770694354 ((RA,Dec)=(83.606129972895,22.0349977948825))  
 2018-01-23T10:38:24: TS value (bin 56) .....: 12782.2401264478 ((RA,Dec)=(83.552189925403,22.0349801539462))  
 2018-01-23T10:38:24: TS value (bin 57) .....: 4894.57471321232 ((RA,Dec)=(83.4982498980654,22.0349448720938))  
 2018-01-23T10:38:25: TS value (bin 58) .....: 1308.81614562467 ((RA,Dec)=(83.4443099043183,22.0348919493651))  
 2018-01-23T10:38:26: TS value (bin 59) .....: 184.176186502184 ((RA,Dec)=(83.3903699575978,22.0348213858204))  
 2018-01-23T10:38:27: TS value (bin 60) .....: 171.077064988262 ((RA,Dec)=(83.8759157120586,22.08482132278))  
 2018-01-23T10:38:28: TS value (bin 61) .....: 967.541192095814 ((RA,Dec)=(83.8219567277709,22.0848919112295))  
 2018-01-23T10:38:28: TS value (bin 62) .....: 3330.12160152892 ((RA,Dec)=(83.7679976963559,22.0849448526368))  
 2018-01-23T10:38:29: TS value (bin 63) .....: 8326.63355780073 ((RA,Dec)=(83.7140386312786,22.0849801469417))  
 2018-01-23T10:38:29: TS value (bin 64) .....: 13296.291553407 ((RA,Dec)=(83.6600795460037,22.0849977941042))  
 2018-01-23T10:38:29: TS value (bin 65) .....: 13268.7162866299 ((RA,Dec)=(83.6061204539964,22.0849977941042))  
 2018-01-23T10:38:30: TS value (bin 66) .....: 8271.44478002837 ((RA,Dec)=(83.5521613687215,22.0849801469417))  
 2018-01-23T10:38:30: TS value (bin 67) .....: 3235.92714631031 ((RA,Dec)=(83.4982023036441,22.0849448526368))  
 2018-01-23T10:38:32: TS value (bin 68) .....: 856.084907855373 ((RA,Dec)=(83.4442432722292,22.0848919112295))  
 2018-01-23T10:38:32: TS value (bin 69) .....: 100.486084928474 ((RA,Dec)=(83.3902842879414,22.08482132278))  
 2018-01-23T10:38:33: TS value (bin 70) .....: 63.5295161910763 ((RA,Dec)=(83.8760014424047,22.1348212596949))  
 2018-01-23T10:38:34: TS value (bin 71) .....: 420.276835247409 ((RA,Dec)=(83.8220234070634,22.1348918730668))  
 2018-01-23T10:38:35: TS value (bin 72) .....: 1512.41682718336 ((RA,Dec)=(83.7680453244941,22.134944833166))  
 2018-01-23T10:38:36: TS value (bin 73) .....: 3501.86398137527 ((RA,Dec)=(83.7140672081903,22.1349801399322))  
 2018-01-23T10:38:36: TS value (bin 74) .....: 5456.21998563007 ((RA,Dec)=(83.6600890716457,22.1349977933254))  
 2018-01-23T10:38:36: TS value (bin 75) .....: 5423.35540353489 ((RA,Dec)=(83.6061109283543,22.1349977933254))  
 2018-01-23T10:38:37: TS value (bin 76) .....: 3415.80120816542 ((RA,Dec)=(83.5521327918098,22.1349801399322))  
 2018-01-23T10:38:39: TS value (bin 77) .....: 1444.62500176855 ((RA,Dec)=(83.4981546755059,22.134944833166))  
 2018-01-23T10:38:40: TS value (bin 78) .....: 347.748621415958 ((RA,Dec)=(83.4441765929366,22.1348918730668))  
 2018-01-23T10:38:41: TS value (bin 79) .....: 28.9982427835639 ((RA,Dec)=(83.3901985575953,22.1348212596949))  
 2018-01-23T10:38:41: TS value (bin 80) .....: 0.000140878517413512 ((RA,Dec)=(83.8760872336359,22.1848211965651))  
 2018-01-23T10:38:42: TS value (bin 81) .....: 107.681400419999 ((RA,Dec)=(83.8220901337113,22.184891834877))  
 2018-01-23T10:38:43: TS value (bin 82) .....: 450.993852517771 ((RA,Dec)=(83.7680929864576,22.1849448136813))  
 2018-01-23T10:38:44: TS value (bin 83) .....: 1070.85951633804 ((RA,Dec)=(83.7140958053972,22.1849801329177))  
 2018-01-23T10:38:45: TS value (bin 84) .....: 1608.07341634043 ((RA,Dec)=(83.6600986040529,22.184997792546))  
 2018-01-23T10:38:46: TS value (bin 85) .....: 1594.21778547196 ((RA,Dec)=(83.6061013959471,22.184997792546))  
 2018-01-23T10:38:49: TS value (bin 86) .....: 1044.80755175877 ((RA,Dec)=(83.5521041946027,22.1849801329177))  
 2018-01-23T10:38:50: TS value (bin 87) .....: 416.251921858202 ((RA,Dec)=(83.4981070135424,22.1849448136813))  
 2018-01-23T10:38:51: TS value (bin 88) .....: 85.6344042030105 ((RA,Dec)=(83.4441098662887,22.184891834877))  
 2018-01-23T10:38:51: TS value (bin 89) .....: 8.92718962859362e-05 ((RA,Dec)=(83.3901127663642,22.1848211965651))  
 2018-01-23T10:38:52: TS value (bin 90) .....: 1.91133585758507e-05 ((RA,Dec)=(83.8761730859477,22.2348211333902))  
 2018-01-23T10:38:53: TS value (bin 91) .....: 0.000157859758473933 ((RA,Dec)=(83.8221569078666,22.23489179666))  
 2018-01-23T10:38:53: TS value (bin 92) .....: 86.4110524116841 ((RA,Dec)=(83.7681406823552,22.2349447941828))  
 2018-01-23T10:38:54: TS value (bin 93) .....: 226.011403005687 ((RA,Dec)=(83.7141244229647,22.2349801258983))  
 2018-01-23T10:38:55: TS value (bin 94) .....: 356.216069200891 ((RA,Dec)=(83.6601081432468,22.2349977917661))  
 2018-01-23T10:38:56: TS value (bin 95) .....: 358.411581601045 ((RA,Dec)=(83.6060918567532,22.2349977917661))  
 2018-01-23T10:38:57: TS value (bin 96) .....: 225.003625298123 ((RA,Dec)=(83.5520755770353,22.2349801258983))  
 2018-01-23T10:38:58: TS value (bin 97) .....: 82.1810223747161 ((RA,Dec)=(83.4980593176448,22.2349447941828))  
 2018-01-23T10:38:59: TS value (bin 98) .....: 0.000161765521625057 ((RA,Dec)=(83.4440430921334,22.23489179666))  
 2018-01-23T10:38:59: TS value (bin 99) .....: 2.75422062259167e-05 ((RA,Dec)=(83.3900269140523,22.2348211333902))

**#22 - 01/23/2018 11:45 AM - Knödseder Jürgen**

- Status changed from *In Progress* to *Closed*

- % Done changed from 90 to 100

Close the feature now.

**Files**

---

tmap.py	749 Bytes	09/27/2016	Mayer Michael
tmap.fits	33.8 KB	09/27/2016	Mayer Michael
tmap.png	33.8 KB	01/23/2018	Knödseder Jürgen
tmap_orig.png	33.9 KB	01/23/2018	Knödseder Jürgen