MATLAB usage

New MATLAB DOOCS interface

- Introduction: reason for searching for new solutions.
- New MATLAB MEX files for communication with DOOCS servers
- Wrapper MATLAB script to make all PITZ MATLAB scripts working normally with new MEX file

MATLAB scripts responsibility

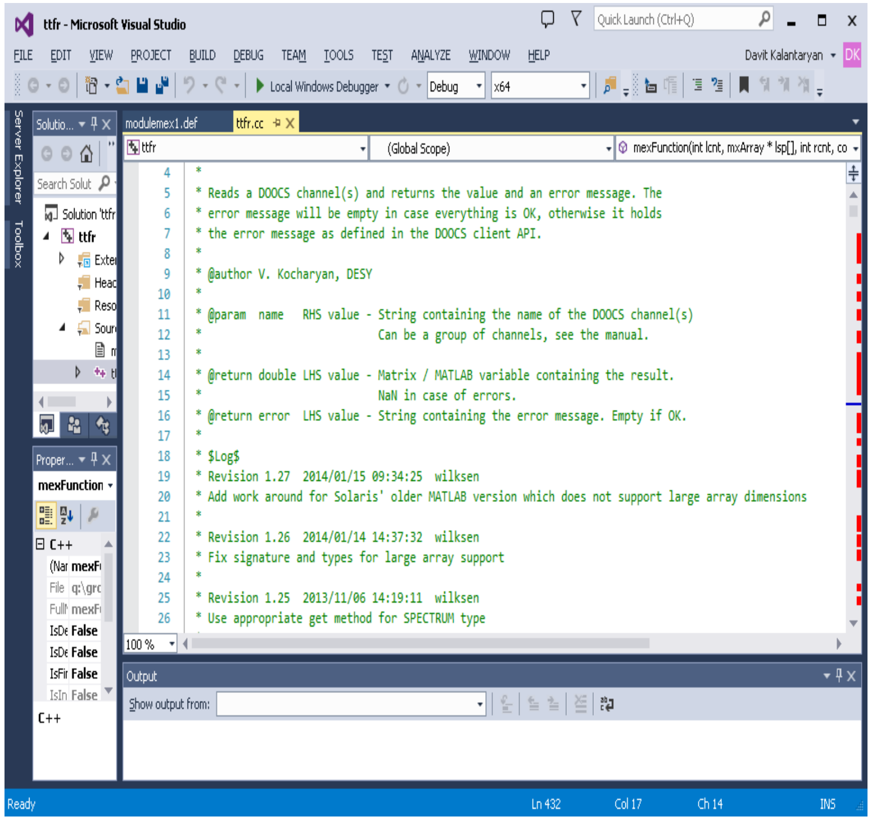
Davit Kalantaryan, Igor Isaev

Introduction: reason for searching for new solutions

> Reasons to exchange **TTFR / TTFW** to **doocsread / doocswrite**

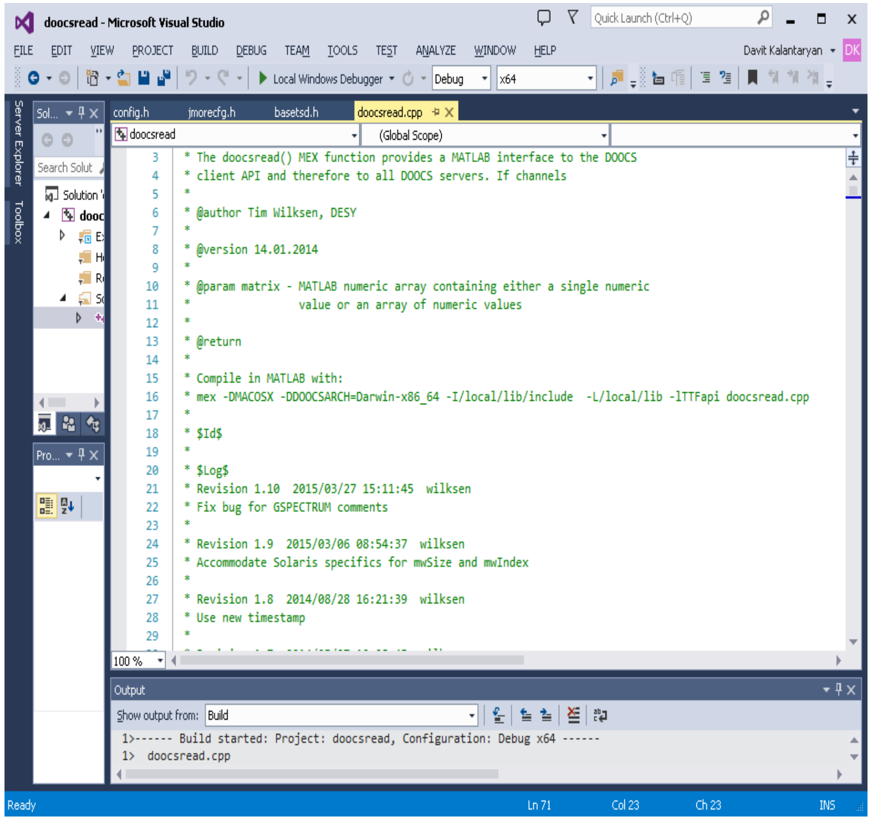
- Possible solution of MATLAB crashes
- TTFW/R is not supported anymore -> could be incorrect function work (such as memory corruption)

TTFR


ttfr - Microsoft Visual Studio
modulemex1.def
ttfr.cc
(Global Scope)
mexFunction(int lcnt, mxArray * lsp[], int rcnt, co
Ready
Ln 432
Col 17
Ch 14
INS

```
 *
 * Reads a DOOCS channel(s) and returns the value and an error message. The
 * error message will be empty in case everything is OK, otherwise it holds
 * the error message as defined in the DOOCS client API.
 *
 * @author V. Kocharyan, DESY
 *
 * @param  name   RHS value - String containing the name of the DOOCS channel(s)
 *                            Can be a group of channels, see the manual.
 *
 * @return double LHS value - Matrix / MATLAB variable containing the result.
 *                            NaN in case of errors.
 * @return error  LHS value - String containing the error message. Empty if OK.
 *
 * $Log$
 * Revision 1.27  2014/01/15 09:34:25  wilksen
 * Add work around for Solaris' older MATLAB version which does not support large array dimensions
 *
 * Revision 1.26  2014/01/14 14:37:32  wilksen
 * Fix signature and types for large array support
 *
 * Revision 1.25  2013/11/06 14:19:11  wilksen
 * Use appropriate get method for SPECTRUM type
```

doocsread



```
 * The doocsread() MEX function provides a MATLAB interface to the DOOCS
 * client API and therefore to all DOOCS servers. If channels
 *
 * @author Tim Wilksen, DESY
 *
 * @version 14.01.2014
 *
 * @param matrix - MATLAB numeric array containing either a single numeric
 *                 value or an array of numeric values
 *
 * @return
 *
 * Compile in MATLAB with:
 * mex -DMACOSX -DDOOCSARCH=Darwin-x86_64 -I/local/lib/include  -L/local/lib -lTTFapi doocsread.cpp
 *
 * $Id$
 *
 * $Log$
 * Revision 1.10  2015/03/27 15:11:45  wilksen
 * Fix bug for GSPECTRUM comments
 *
 * Revision 1.9  2015/03/06 08:54:37  wilksen
 * Accommodate Solaris specifics for mwSize and mwIndex
 *
 * Revision 1.8  2014/08/28 16:21:39  wilksen
 * Use new timestamp
 *
```

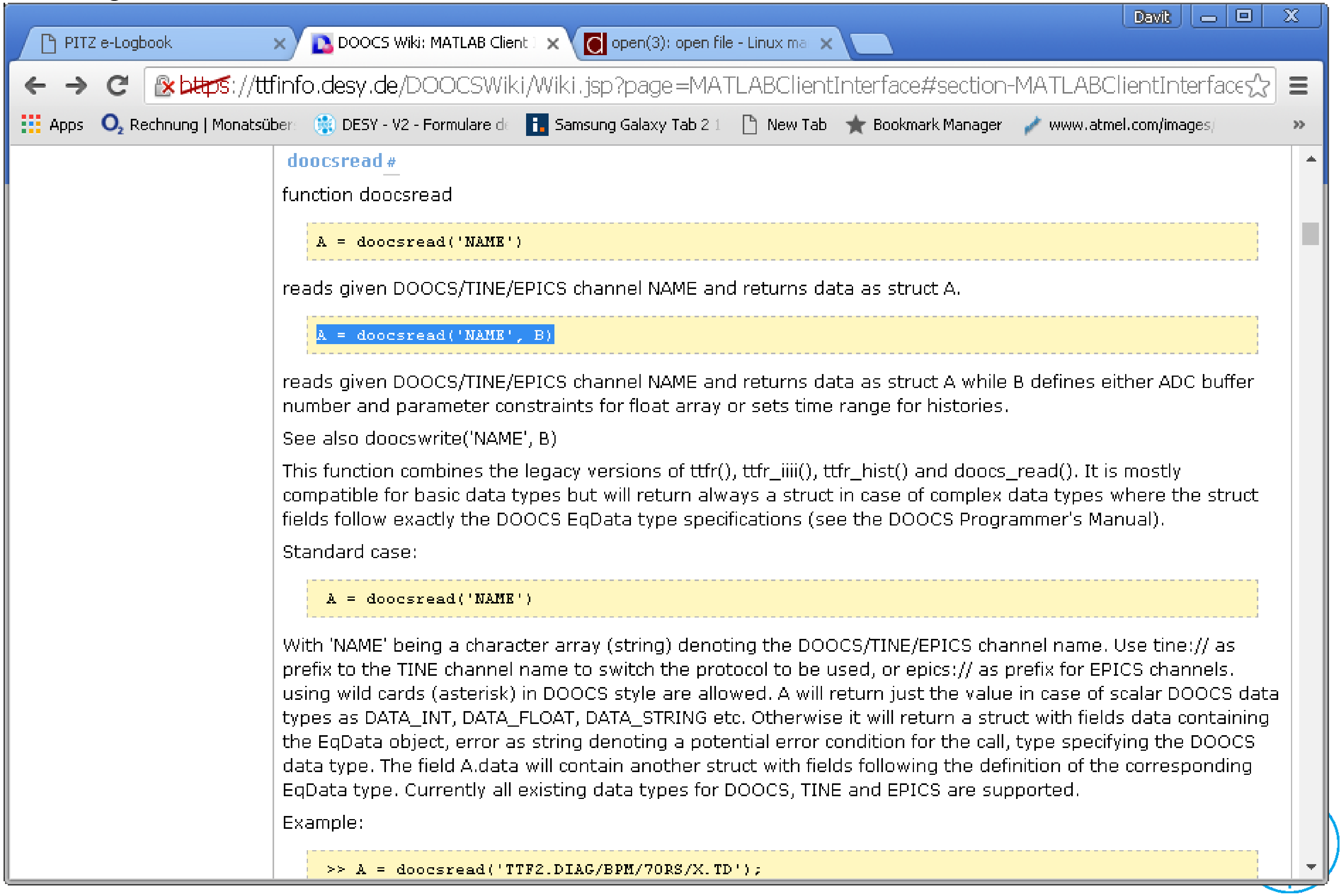
```
1>------ Build started: Project: doocsread, Configuration: Debug x64 ------
1>  doocsread.cpp
```

Introduction: reason for searching for new solutions 2

- Starting **from 2014** January **maintenance** of ‘ttfr’, ‘ttfw’ was **stopped**.
- Old functions (ttfr,ttfw, ttfr_iiii, ttfr_hist, doocs_read) **do not work with newer DOOCS servers**, specially with MTCA servers, specially DOOCS history reading functionality
- **Functionality** of doocsread and doocswrite is more **richer**. For example during the ‘doocread’ call one can provide some data to server. For example history start and end time for getting history from server, or to provide event number to timing service, for getting corresponding time from server
- There is **recommendation** from last maintainer of ‘ttfr’, ‘ttfw’ and author and maintainer of ‘doocsread’, ‘doocswrite’ Tim Wilksen to use these **new functions** instead of old DOOCS MATLAB tools.

‘doocsread’ for reading data from DOOCS servers

1. http://tesla.desy.de/doocs/doocs.html
2. Programmer's Manual
3. MATLAB Client Interface
4. doocsread


PITZ e-Logbook
DOOCS Wiki: MATLAB Client
open(3): open file - Linux ma
Davit
https://ttfinfo.desy.de/DOOCSWiki/Wiki.jsp?page=MATLABClientInterface#section-MATLABClientInterface
Apps
Rechnung | Monatsüber
DESY - V2 - Formulare d
Samsung Galaxy Tab 2 1
New Tab
Bookmark Manager
www.atmel.com/images/

doocsread #

function doocsread

```
A = doocsread('NAME')
```

reads given DOOCS/TINE/EPICS channel NAME and returns data as struct A.

```
A = doocsread('NAME', B)
```

reads given DOOCS/TINE/EPICS channel NAME and returns data as struct A while B defines either ADC buffer number and parameter constraints for float array or sets time range for histories.

See also doocswrite('NAME', B)

This function combines the legacy versions of ttfr(), ttfr_iiii(), ttfr_hist() and doocs_read(). It is mostly compatible for basic data types but will return always a struct in case of complex data types where the struct fields follow exactly the DOOCS EqData type specifications (see the DOOCS Programmer's Manual).

Standard case:

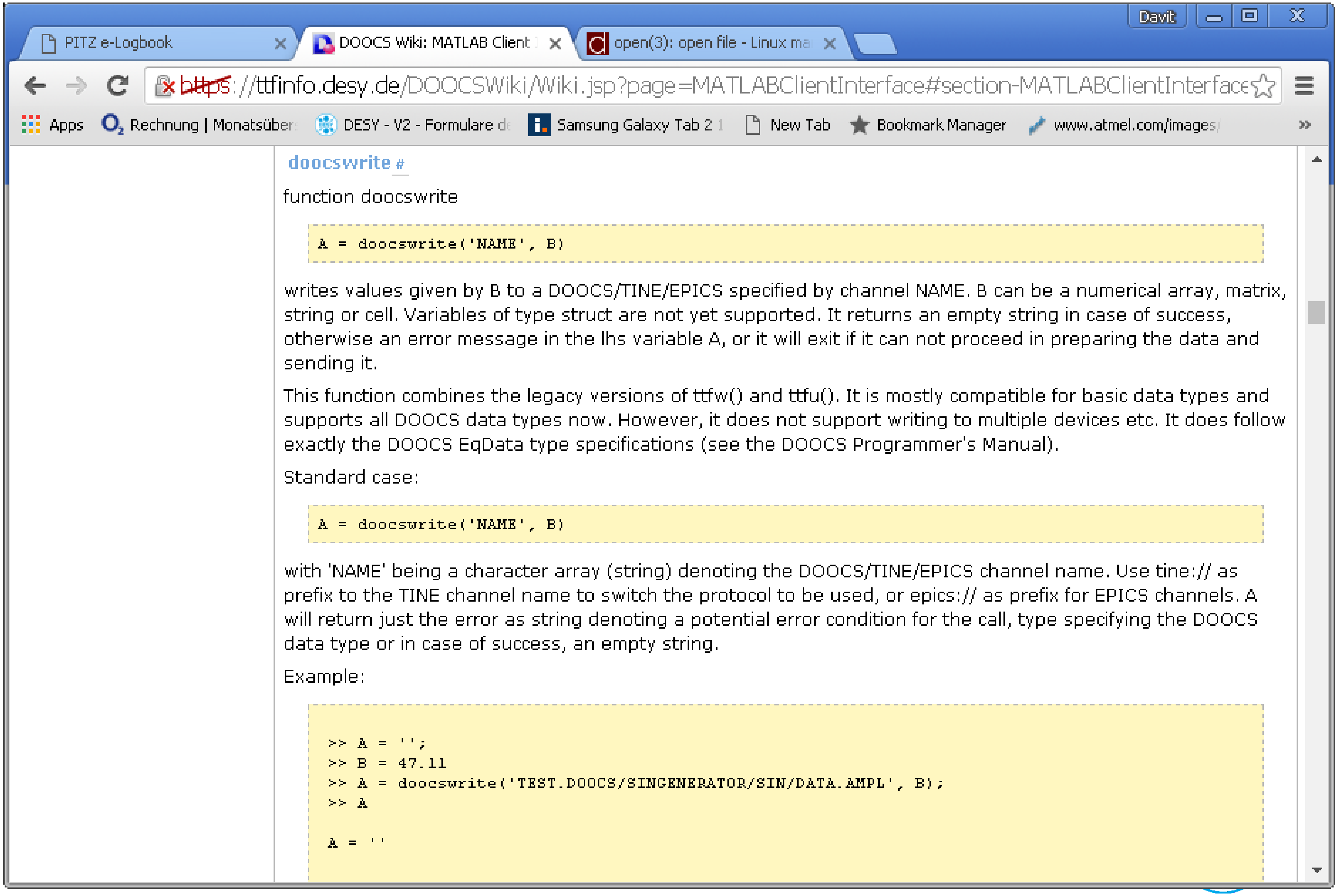
```
 A = doocsread('NAME')
```

With 'NAME' being a character array (string) denoting the DOOCS/TINE/EPICS channel name. Use tine:// as prefix to the TINE channel name to switch the protocol to be used, or epics:// as prefix for EPICS channels. using wild cards (asterisk) in DOOCS style are allowed. A will return just the value in case of scalar DOOCS data types as DATA_INT, DATA_FLOAT, DATA_STRING etc. Otherwise it will return a struct with fields data containing the EqData object, error as string denoting a potential error condition for the call, type specifying the DOOCS data type. The field A.data will contain another struct with fields following the definition of the corresponding EqData type. Currently all existing data types for DOOCS, TINE and EPICS are supported.

Example:

```
 >> A = doocsread('TTF2.DIAG/BPM/7ORS/X.TD');
```

‘doocswrite’ for writing data to DOOCS servers


PITZ e-Logbook
DOOCS Wiki: MATLAB Client
open(3): open file - Linux ma
~~https~~://ttfinfo.desy.de/DOOCSWiki/Wiki.jsp?page=MATLABClientInterface#section-MATLABClientInterface
Apps
Rechnung | Monatsüber
DESY - V2 - Formulare d
Samsung Galaxy Tab 2
New Tab
Bookmark Manager
www.atmel.com/images

doocswrite #

function doocswrite

```
A = doocswrite('NAME', B)
```

writes values given by B to a DOOCS/TINE/EPICS specified by channel NAME. B can be a numerical array, matrix, string or cell. Variables of type struct are not yet supported. It returns an empty string in case of success, otherwise an error message in the lhs variable A, or it will exit if it can not proceed in preparing the data and sending it.

This function combines the legacy versions of ttfw() and ttfu(). It is mostly compatible for basic data types and supports all DOOCS data types now. However, it does not support writing to multiple devices etc. It does follow exactly the DOOCS EqData type specifications (see the DOOCS Programmer's Manual).

Standard case:

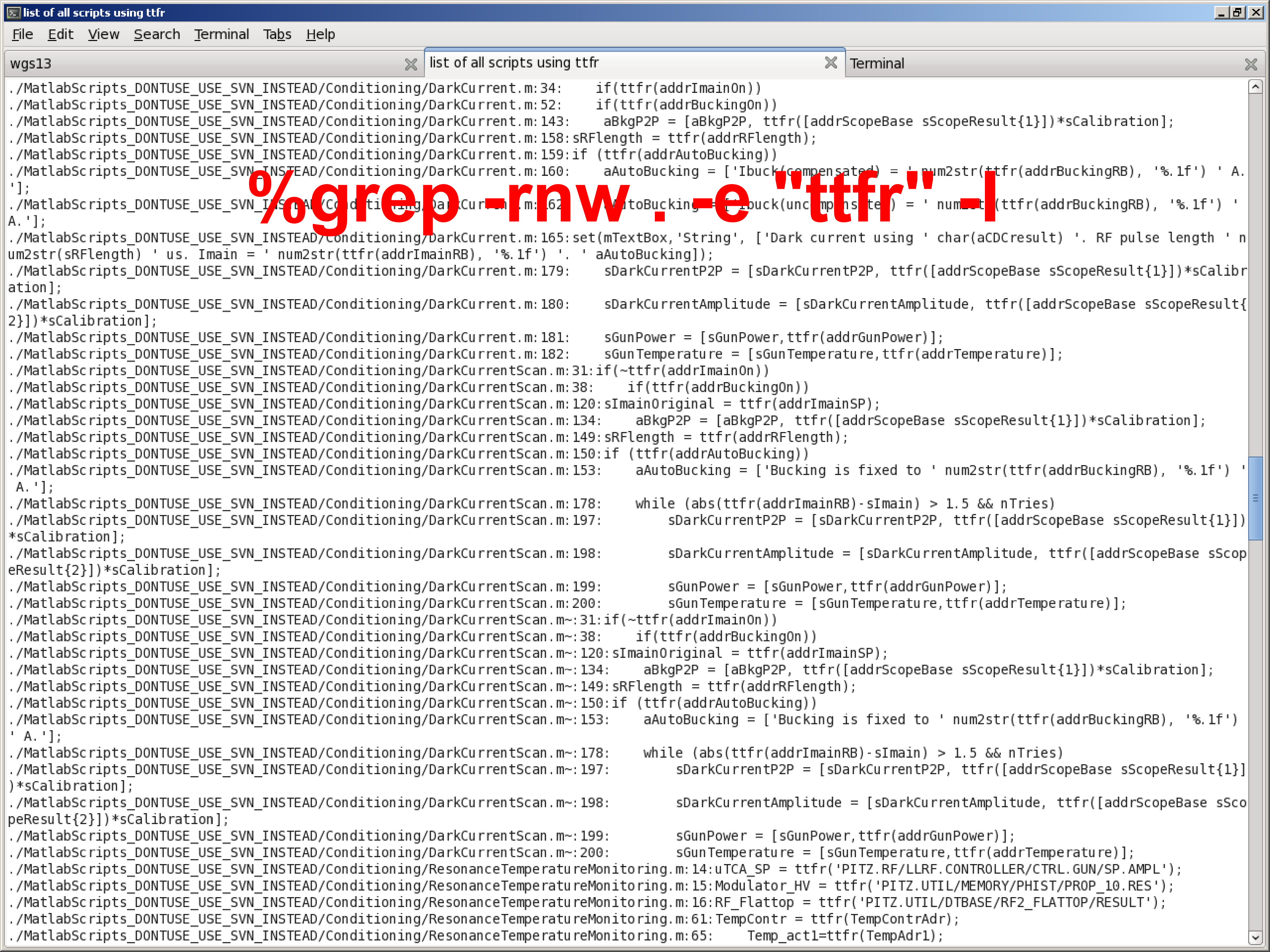
```
A = doocswrite('NAME', B)
```

with 'NAME' being a character array (string) denoting the DOOCS/TINE/EPICS channel name. Use tine:// as prefix to the TINE channel name to switch the protocol to be used, or epics:// as prefix for EPICS channels. A will return just the error as string denoting a potential error condition for the call, type specifying the DOOCS data type or in case of success, an empty string.

Example:

```
>> A = '';
>> B = 47.11
>> A = doocswrite('TEST.DOOCS/SINGENERATOR/SIN/DATA.AMPL', B);
>> A

A = ''
```


list of all scripts using ttfr
File Edit View Search Terminal Tabs Help
wgs13
list of all scripts using ttfr
Terminal

%grep -rnw . -e "ttfr" -l

```
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:34:    if(ttfr(addrImainOn))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:52:    if(ttfr(addrBuckingOn))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:143:    aBkgP2P = [aBkgP2P, ttfr([addrScopeBase sScopeResult{1}])*sCalibration];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:158:sRFlength = ttfr(addrRFlength);
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:159:if (ttfr(addrAutoBucking))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:160:    aAutoBucking = ['Ibuck(compensated) = ' num2str(ttfr(addrBuckingRB), '%.1f') ' A.
'];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:162:    aAutoBucking = ['Ibuck(uncompensated) = ' num2str(ttfr(addrBuckingRB), '%.1f') '
A.'];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:165:set(mTextBox,'String', ['Dark current using ' char(aCDCresult) '. RF pulse length ' n
um2str(sRFlength) ' us. Imain = ' num2str(ttfr(addrImainRB), '%.1f') '. ' aAutoBucking]);
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:179:    sDarkCurrentP2P = [sDarkCurrentP2P, ttfr([addrScopeBase sScopeResult{1}])*sCalibr
ation];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:180:    sDarkCurrentAmplitude = [sDarkCurrentAmplitude, ttfr([addrScopeBase sScopeResult{
2}])*sCalibration];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:181:    sGunPower = [sGunPower,ttfr(addrGunPower)];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrent.m:182:    sGunTemperature = [sGunTemperature,ttfr(addrTemperature)];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:31:if(~ttfr(addrImainOn))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:38:    if(ttfr(addrBuckingOn))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:120:sImainOriginal = ttfr(addrImainSP);
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:134:    aBkgP2P = [aBkgP2P, ttfr([addrScopeBase sScopeResult{1}])*sCalibration];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:149:sRFlength = ttfr(addrRFlength);
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:150:if (ttfr(addrAutoBucking))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:153:    aAutoBucking = ['Bucking is fixed to ' num2str(ttfr(addrBuckingRB), '%.1f') '
 A.'];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:178:    while (abs(ttfr(addrImainRB)-sImain) > 1.5 && nTries)
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:197:        sDarkCurrentP2P = [sDarkCurrentP2P, ttfr([addrScopeBase sScopeResult{1}])
*sCalibration];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:198:        sDarkCurrentAmplitude = [sDarkCurrentAmplitude, ttfr([addrScopeBase sScop
eResult{2}])*sCalibration];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:199:        sGunPower = [sGunPower,ttfr(addrGunPower)];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m:200:        sGunTemperature = [sGunTemperature,ttfr(addrTemperature)];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:31:if(~ttfr(addrImainOn))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:38:    if(ttfr(addrBuckingOn))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:120:sImainOriginal = ttfr(addrImainSP);
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:134:    aBkgP2P = [aBkgP2P, ttfr([addrScopeBase sScopeResult{1}])*sCalibration];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:149:sRFlength = ttfr(addrRFlength);
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:150:if (ttfr(addrAutoBucking))
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:153:    aAutoBucking = ['Bucking is fixed to ' num2str(ttfr(addrBuckingRB), '%.1f')
' A.'];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:178:    while (abs(ttfr(addrImainRB)-sImain) > 1.5 && nTries)
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:197:        sDarkCurrentP2P = [sDarkCurrentP2P, ttfr([addrScopeBase sScopeResult{1}]
)*sCalibration];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:198:        sDarkCurrentAmplitude = [sDarkCurrentAmplitude, ttfr([addrScopeBase sSco
peResult{2}])*sCalibration];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:199:        sGunPower = [sGunPower,ttfr(addrGunPower)];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/DarkCurrentScan.m~:200:        sGunTemperature = [sGunTemperature,ttfr(addrTemperature)];
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/ResonanceTemperatureMonitoring.m:14:uTCA_SP = ttfr('PITZ.RF/LLRF.CONTROLLER/CTRL.GUN/SP.AMPL');
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/ResonanceTemperatureMonitoring.m:15:Modulator_HV = ttfr('PITZ.UTIL/MEMORY/PHIST/PROP_10.RES');
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/ResonanceTemperatureMonitoring.m:16:RF_Flattop = ttfr('PITZ.UTIL/DTBASE/RF2_FLATTOP/RESULT');
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/ResonanceTemperatureMonitoring.m:61:TempContr = ttfr(TempContrAdr);
./MatlabScripts_DONTUSE_USE_SVN_INSTEAD/Conditioning/ResonanceTemperatureMonitoring.m:65:    Temp_act1=ttfr(TempAdr1);
```

Wrapper MATLAB script to make all PITZ MATLAB scripts working normally with new MEX file

- There are a **lot of MATLAB scripts** that use ‘**ttfr**’, ‘**ttfw**’.
- Modifying all of them will take a lot of time.
- As a **workaround** will be created ‘**ttfr.m**’ and ‘**ttfw.m**’ MATLAB scripts, those under hood use ‘doocsread’ and ‘doocswrite’. So all scripts, those use these functions, will work.
- Script will be something like this:

```
==================================================================
% ttfr.m
function [value error_string]=ttf(doocs_address)

disp(‘WARNING:    ttfr should be exchanged by doocsread !!!’);
doocs_value_str = doocsread(doocs_address);
value = doocs_value_str.data;
error_string = doocs_value_str.error;
==================================================================
```

MATLAB script/tool name	Responsible person
TDS measurement	Holger
Long ph sp (Malyutin)	Houjun, Mikhail
BBA	Yves
Phase stability measurements	Igor
QE	**Tino?**
QE map	**Tino?**
Charge measurement	Mikhail
LILI	Igor, Davit
Phase scan GUI	Mikhail
LT scan	Mikhail
DC scan	Mikhail
Mirror 56	**Tino?**
OMA	Holger
OTE tool	**Tino?**
Resonance Temperature Monitor	Yves
uTCA plot evaluator	Mikhail
Charge solenoid scan	Mikhail
Trajectory and booster steering (from Marek)	**Ye ?**
Solenoid BBA	Mikhail

Other script/tool name	Responsible person
SLEM	
Tomography	
EmWIZ	Stefan
SMAC	David
Video client	Stefan