

---

Triangle MicroWorks, Inc.  
IEC 60870-5-102 Master  
Source Code Library

What's New?

*a description of the changes included in all versions of the library*

Version 3.09.00  
March 15, 2010

---

Property of Triangle MicroWorks, Inc.



This Source Code and the associated Documentation contain proprietary information of Triangle MicroWorks, Inc. and may not be copied or distributed in any form without the written permission of Triangle MicroWorks, Inc.

Copies of the source code may be made only for backup purposes.






© 1994 - 2010 Triangle MicroWorks, Inc. All rights reserved.

This document describes features or corrections that have been added to the IEC 60870-5-101 Master Source Code Library.


The symbols to the left of each revision are used to help define the following kinds of revisions:

- 101 Additional IEC 60870-5-101 Features, giving the Library support for more IEC 60870-5-101 objects or functions.
-  Additional Implementation Features, allowing implementers to more efficiently install the Source Code Library.
-  Corrections to problems, with indications for when the problems were introduced.

### **Version 3.09.00 (March 15, 2010)**

-  Added user data pointers to TMWCHNL, TMWSCTR, and TMWSESN.
-  *Description:* When multiple application contexts are used, closing one deleted the memory pool lock, which could cause problems if another application context was open.  
*Introduced:* v3.00  
*Resolution:* Fixed in v3.09
-  *Description:* WinIOTarg could fail to process all available characters, causing timeouts when running at high baud rates.  
*Introduced:* v3.00  
*Resolution:* Fixed in v3.09
-  *Description:* When configured to use system time, WinIOTarg System time not setting the dstInEffect flag correctly.  
*Introduced:* v3.00  
*Resolution:* Fixed in v3.09
-  *Description:* LinIOTarg could cause high levels of processor utilization when multiple channels were opened.  
*Introduced:* v3.07  
*Resolution:* Fixed in v3.09

### **Version 3.08.00 (September 25, 2009)**

-  *Description:* Modifying IEC 60870-5-102 Common Address of ASDU Size did not work.  
*Introduced:* v3.00  
*Resolution:* Fixed in v3.08



*Description:* The rs232Reader thread in WinIOTarg was missing a stop event.  
*Introduced:* v3.00  
*Resolution:* Fixed in v3.08

### **Version 3.07.00 (July 17, 2009)**



Added LinIOTarg (low-level target for Linux) and new command line-based examples (that can be used with Linux).



Integrated WinIOTarg (low-level target for Windows) into the standard release package.



Added FormatMessage Win32 API function to WinIOTarg to provide more descriptive errors instead of just numeric values.

### **Version 3.06.00 (April 14, 2009)**



Added TMWCNFG\_MAX\_APPLRCVS to “break out” of loop in tmwappl\_checkForInput() after the specified number of iterations.



Modified timer values in tmwlink\_channelCallback to speed up retries.

### **Version 3.05.00 (December 4, 2008)**

No updates to the IEC 60870-5-102 Master Source Code Library in this release.

### **Version 3.04.01 (September 30, 2008)**



Improved comments on compiler options such as TMWCONFIG\_USE\_MANAGED\_SCL.

### **Version 3.04 (September 12, 2008)**



*Description:* tmwdb\_lockQueue and tmwdb\_unlockQueue were described in the header comments for tmwdb\_storeEntry in tmwdb.h, but they were not defined in tmwdb.c.

*Introduced:* v3.00

*Resolution:* Fixed in v3.04

### **Version 3.03.00 (April 1, 2008)**



*Description:* The Source Code Library did not poll properly for Class 1 data if class1PollCount ==0 and only one session was configured.

*Introduced:* v3.00

*Resolution:* Fixed in v3.04



*Description:* tmwdb\_lockQueue and tmwdb\_unlockQueue were described in the header comments for tmwdb\_storeEntry in tmwdb.h, but they were not defined in tmwdb.c.

*Introduced:* v3.00

*Resolution:* Fixed in v3.04

### **Version 3.02.00 (December 7, 2007)**

No updates to the IEC 60870-5-102 Master Source Code Library in this release.

### **Version 3.01.01 (August 8, 2007)**



Added ability to keep time on a per-session basis. Added pSession to tmwvertime structure; target layer can use this parameter to return a time specific to the session.

### **Version 3.01.00 (May 23, 2007)**






Initial release of IEC 60870-5-102 v3.x Master Source Code Library.

The Triangle MicroWorks, Inc. Source Code architecture was redesigned and reimplemented to follow modern software engineering practices and leverage new techniques in software design. We have also incorporated suggestions from our existing customer base. The main advantages to this redesign include:

- 1) Common source code architecture across all TMW libraries - The use of a common architecture across all libraries significantly reduces time required to port additional protocols to a target device. In addition, using common software across all products results in better code, since common functions are utilized much more frequently than protocol specific functions.
- 2) Source Code Library calling routines are now compatible with a wider variety of event driven techniques, allowing the target application to be achieve higher performance.
- 3) Most configuration parameters are now passed as arguments in Source Code Library function calls instead of macros. This allows for data hiding and protects configuration parameters from accidentally being changed while the Source Code Library is running. In addition, macro calls to target hardware and the database interface routines were replaced with function calls.
- 4) Significant reduction in the time to implement new features - In addition to being more flexible, the use of modern software design practices significantly reduces the time required for Triangle MicroWorks to support new features. As the existing protocols are constantly being improved by the associated Technical Committees and Working Groups, it is essential that the Source Code Libraries are able to remain up to date with the latest standards.



### **Version 2.14 (July 10, 2002):**

-  Removed `M101CNFG_LINK_SUPPORT_BALANCED()` and `M101CNFG_LINK_SUPPORT_UNBALANCED()` from `m101cnfg.h`. Both are supported at all times now.
-  Fixed a bug with broadcast messages which contained a time stamp. The time stamp was added at the time the message was created, not when the message was transmitted through callbacks. Now the timestamp is added at the time of transmission and is more accurate.
-  Fixed bugs in file transfer which caused the segment size to be hardcoded to 230 bytes, the information object address to be hardcoded to zero and the checksum in the last segment request to be incorrect.



### **Version 2.13 (March 21, 2002):**

- 101** Added support for protection equipment events including ASDU types 17(MEPTA), 18(MEPTB), 19(MEPTC), 38(MEPTD), 39(MEPT E), and 40(MEPTF).

### **Version 2.12 (January 24, 2002):**

- 101** Added support for File Transfer, this includes support for file transfer in both the monitor and control directions..
- 101** Added functionality to immediately try to reestablish a connection to devices that are offline when a new application level request is issued to that device. This only effects unbalanced communications channels.
-  Added functionality to cancel link level requests if the application level times out. The link layer should not continue to try and send a request if the application layer has timed out.
-  Fixed balanced mode only compilation issues.


### **Version 2.11 (December 11, 2001):**

-  Fixed a bug that was keeping the DIR bit from being set correctly when operating in balanced mode.
- 101** The `M101CNFG_LINK_CLASS1/2_POLL_DLY` parameters for a given channel where being applied in sequence as opposed to independently. Specifically, if two sessions were assigned to a single channel each session would be polled at a rate equal to the sum of the polling periods for each session as opposed to each session's specific polling period. This is now fixed so that each session is polled at its configured polling period independent of other sessions on that channel.
-  Fixed a bug that caused an unbalanced master to delay polling a slave for the response to a request until the end of the current polling period. The master now immediately starts polling a session with a request pending at the rate determined by the `M101CNFG_LINK_CLASS1/2_PENDING_DLY` when a request is sent.









### Version 2.10 (December 4, 2001):




-  Miscellaneous bug fixes in balanced mode support.

### Version 2.09 (November 9, 2001):





- 101 Redesigned initialization of balanced mode to be more robust and added options to support inhibiting user data transmit until the initializations of both ends of the communications channel are complete.
- 101 Added support for custom configuration table ASDU types (MCTNA and CCTNA).
- 101 Added timestamps to protocol analyzer display.
-  Fixed a bug that was causing the DIR bit to be set for unbalanced mode transmissions. This bit is marked 'reserved' for unbalanced mode and should be 0.







### Version 2.08 (September 4, 2001):

- 101 Added support for m101 test command (Type ID **C\_TS\_NA\_1**).
- 101 Added new macros: **M101TARG\_MEMCOPY()** to specify local memory copy implementation. **M101DATA\_SESSION\_NOTIFY\_ONLINE()** to support target notification of online/offline events. **M101TARG\_MALLOC()** and **M101TARG\_FREE()** to support dynamic allocation of memory. **M101CNFG\_OFFLINE\_POLL\_DLY()** to specify how often to poll a slave device if it is detected to be offline.
- 101 Added support for custom ASDU types: 240 – Integrated Totals BCD, 241 – Integrated Totals BDC with Time Tag and 242 – Integrated Totals BDC Command.
- 101 Added limited file ID support (Type ID's 122 and 123).
-  Renamed **tmwphys\_initChannel()** to **tmwphys\_openChannel()** which now calls **M101TARG\_COMM\_OPEN()**.
-  Added new module **m101poll** with functionality from **m101link** and **i870link1**.
-  Renamed **M101LINK\_FLAGS\_xxxx** to **M101TASK\_TX\_FLAGS\_xxxx**.
-  Added new doubly linked list utilities in **tmwdblink.c** and **tmwdblink.h**.
-  Changed parameters to **I870SCL\_PROCESS\_INCOMING\_FUNC**.
-  Changed application callback functions to be accessed through a structure to reduce the number of calling parameters to **I870SCL\_BUILD\_OUTGOING\_FUNC**. Added **beforeTxCallback** and **callbackParam** to application callback capabilities.
-  Added **TMWSTAT\_LINK\_UNEXPECTED\_ACK** and **TMWSTAT\_LINK\_REQUEST\_CANCELLED** error codes.
-  Added **pTransmitStartAsdu** to **M101APPL\_TRANS\_DATA** for use in callback functions that store time immediately before transmission. This fixed problem in version 2.06 that did not store time correctly in such messages.

-  Corrected a problem with `tmwclprm_parseSector()` that could have resulted in illegal references of memory when multiple Source Code Libraries are installed. Introduced in version 2.05.
-  Fixed compile time errors when `M101CNFG_LINK_SUPPORT_UNBALANCD()` is set false or when `M101TARG_DIAG_FEATURE_MASK()` is set to zero. Introduced in version 2.05.
-  Corrected bugs with running multiple sessions and sectors on the same communication channel. Introduced in version 2.05.

### **Version 2.06 (March 26, 2001):**





- 101** Diagnostic support added for “Error” Causes of Transmission <44>..<47> (Unknown Type ID, Unknown COT, Unknown ASDU Address and Unknown Information Object Address) defined in 2<sup>nd</sup> Amendment to IEC 60870-5-101 and the COT “Test” flag.
- 101** Added support for 2-octet COT, introducing the `M101CNFG_ASDU_ORIGINATOR_ADDR()` macro to specify the value of the originator address when 2-octet COT is used.
- 101** Added support for the time-tagged control types defined in IEC 60870-5-104. These Type Ids <58>..<64> correspond to Type Ids <45>..<51> with the addition of a 56-bit time tag containing the time at which the command was issued. The macro `M101CNFG_APPL_TIMETAG_COMMANDS()` specifies if the time-tagged or non-time-tagged command type should be used. This option is included to simplify implementations of protocol converters between IEC 60870-5-101 and 60870-5-104.
- 101** Added support for Bitstring-32 objects in the monitor and command directions (Type IDs <7>, <8>, <33>, <51> and <64>). The `M101DATA_STOR_101_MBO()` macro defines the function to store monitored data in the database. The library function `m101brm_send101commandBitstring()` is called to issue a Bitstring-32 command.
- 101** IEC 60870-5-101 2<sup>nd</sup> Addendum specifies that time tags in messages sent by the controlling station are not corrected for transmission delay. Any correction for transmission delay is to be made in the controlled station. The Master SCL no longer performs transmission delay correction on time tags.
- 101** Added the function `m101brm_send101LoadDelay()` to send the time correction delay to a controlled station.
-  Renamed `M101SCL_STAT_XXXX()` error codes to be `TMWSTAT_XXXX()`. These are now shared between several Source Code Libraries.
-  `M101CNFG_SESSION_LINK_ADDRESS()` renamed to `M101CNFG_LINK_ADDRESS()` as this is the name by which this parameter is more commonly known.
-  New macro `M101CNFG_SESSION_IS_ON_LINE()` selects a session’s initial state.
-  Removed unused fields from structure `M870DB1_OBJ_INFO_TABLE` to reduce code space.

-  Database storage functions now return a status code that is eventually returned to the target application in the return status of `m101task_main()`.
-  Added `M101TARG_XXXX_INSTALLED()` macros to identify combinations of protocols that are loaded simultaneously. These are used internally in libraries containing combinations of protocols.
-  Macro `M101TARG_COMM_OPEN()` modified to add parameters to specify a function to be called if the master identifies that the slave has closed the channel.
-  Corrected an error that prevented the ASDU address from being displayed on the diagnostic output. This error was introduced in version 2.05.
-  Corrected an error that set an incorrect 24-bit time value in the `C_CD_NA_1` Delay Acquisition command (Type ID 106).
-  Corrected an error that continued to check confirm timeouts and response timeouts for sessions that were off-line. This could lead to an invalid pointer access.





**Version 2.05 (January 24, 2001):**

- 101 Added support for read command and reset process command (ASDU types 102 and 105, respectively)
- 101 Added support for 56-bit time ASDU types in received monitored messages. More specifically, added support for ASDU types 30 through 37.
- 101 Added support for parameter loading and parameters in the control direction. More specifically, added support for ASDU types 110 through 112.
- 101 Now, depending upon the value of `M101CNFG_APPL_AUTO_MODE_MASK()`, clock synchronization and general interrogation commands may be automatically initiated when the end-of-init ASDU is received from remote devices.
- 101 The link-layer polling algorithm is now configurable to support a maximum number of consecutive polls to a single device as well as a configurable delay after each poll. This can be used to avoid excessive load on the target processor or communication media.
- 101 Added `M101CNFG_SESSION_POLL_ENBLD()` so end user can take remote device offline (no polling for class 1 or 2 data).
- 101 Changed application layer confirmation timeout to be measured from the absolute time the message was transmitted instead of restarted when a non-cyclic data message is received.
- 101 Added macro for `M101DATA_SESSION_NOTIFY_ONLINE()` when communications are lost or restored so that points in Target Application database can be marked accordingly. Replaced performance statistics counters with calls of other `M101DATA_XXXX_NOTIFY_XXXX()` macros.
- 101 Changed duplicate request checking to only check the application layer portion of messages (not the link layer headers).








## Changed filenames and added new files:

-  Renamed almost every core Source Code Library file, some were removed, some were renamed completely. All Source Code Library files that are specific to the slave IEC 608780-5-101 operation now begin with **m101** (instead of **i1m**). Other files, which are not specific to just Slave implementations of IEC 60870-5-101 but are specific to IEC 60870 protocol profiles, have been renamed to use the **i870** filename prefix. These changes were made to allow stronger consistency across the entire Triangle MicroWorks, Inc. Source Code Library product line, and to increase readability.
-  The example database is now entirely contained within **simdata.c** and **simdata.h** modules. This consolidation make it much easier to install the Target Application database by simply editing **m101data.h** to replace references of **simdata** with references to the Target Application Database. After this is done, the **simdata** module can be removed.
-  Added several optional modules that are used to provide a simple command line user-interface. If not needed, this feature can be disabled, and these optional modules can be removed.
-  Added **m101cfts** module, which contains code to check the consistency of configuration parameters.


## TA-to-SCL interface changes:

-  The **i1mdbas\_processQue()** function is no longer required to be called by Target Application software. This function is now called by the Source Code Library through **m101task\_main()** based on a timer configured by **M101CNFG\_PROCESS\_DBAS\_PRD()**. However, if desired, the function, which has been renamed to **m101db1\_processDbas()**, can be called by the Target Application (perhaps within a separate task) if **M101CNFG\_PROCESS\_DBAS\_PRD()** is set to zero.
-  The **i1mbrm.c** file (now **m101brm.c**) no longer needs to be edited (nor should it be edited) in order to generate request messages to be transmitted to remote devices. Now, using functions defined within **m101brm.c** and prototyped in **m101brm.h**, Target Application software can generate completely configurable request messages through simple function calls (Source Code Library TA-to-SCL entry points).
-  With a single function call, request messages can now be configured to be periodic. For example a single function call can initiate a periodic general interrogation request.
-  Callback functions, which are called when the status of a request is known (e.g., successful response-received, failed response-received, time-out, etc.) now report the status through **m101SCL\_STAT** codes, and allow Target Application software to resubmit the request, release (finish) the request, or continue the request (continue to wait for more response data).


## New Target Hardware Interface macros:

-  Added more parameters to `M101TARG_COMM_RECEIVE()` to allow more efficient implementations of the receive operation in the future.
-  Moved and renamed `M101CNFG_PHYS_NUM_COM_CHANNELS()` to `M101TARG_COMM_MAX_NUM_CHNLS()` to further consolidate target-hardware-specific interface parameters.
-  Added `causeOfInit` to `M101TARG_COMM_OPEN()` and `M101TARG_COMM_CLOSE()`.
-  Added `M101TARG_COMM_CHNL_NAME()` to allow diagnostic or protocol analyzer display functions to indicate communication channels using names rather than indices.
-  Added `M101TARG_RESOURCE_LOCK()` and `M101TARG_RESOURCE_UNLOCK()` to support multiple thread operation of Source Code Library.
-  Added `M101TARG_DIAG_FEATURE_MASK()` to enable or disable diagnostic features at compile-time. Disabling features will decrease code-space requirements, and may slightly decrease execution time.
-  Removed `M101TARG_DIAG_INIT_ERROR()`; it is no longer needed because `m101task_init()` returns an error code instead of a Boolean.




## Database Interface changes:

-  The Target Application Database Interface is now entirely encapsulated in `m101data.h` using C-Macros. The `i1mdbas1.c` file (now `m101db1.c`) no longer needs to be edited (nor should it be edited).




## Configuration Interface Changes:

-  Added many new configuration parameters, removed some, and renamed others. The Configuration Interface is now entirely encapsulated in `m101cnfg.h` using C-Macros. The `i1mdvrs.c` file (now `m101dvrs.c`) no longer needs to be edited (nor should it be edited) to set configuration parameters for communication sessions.


## General changes

-  Changed format of diagnostic protocol analyzer message to include communication channel name (through new `I1STARG_COMM_CHNL_NAME()` macro) rather than index, and to more closely match the format of other Triangle MicroWorks, Inc. Source Code Library protocol analyzer displays.
-  With the removal of `I1SCONFG_DIAG_ID_AUX_MASK()`, the filtering of the diagnostic protocol analyzer messages is now totally up to the Target Application.
-  To prevent warnings emitted by specific strict compilers, rearranged code, changed some data types, added example `#pragma`'s, initialized local variables, and added type-casts.






### **Version 1.31 (November 24, 1999):**

- 101 Added support for balanced transmission mode, creating the configuration macros `I1MCONFIG_SUPPORT_BAL()`, `I1MCONFIG_LINK_BALANCED()`, `I1MCONFIG_LINK_DIR()` and `I1MCONFIG_OPT_ONE_CHAR_ACK()` to control this. The support code for balanced mode may be conditionally compiled, and excluded to save code space if not required.
-  Moved definitions for IEC FT1.2 constructs from several files into the single file `IECFT12.h`, with corresponding changes to macro names. This file is shared with other TMW IEC products.
-  Continued revision of source code comments to enhance or more accurately document source code files.
-  Fixed a problem that occurred when using 3-octet Information Object Addresses with a Motorola processor. The octet alignment was incorrect so that 3-octet IOA values were divided by 256 when written to the message stream. This fault has been present since version 1.08 (the initial release of the IEC 60870-5-101 Master Source Code Library).

### **Version 1.29 (June 23, 1999):**

-  Now uses `TMWDEFS_COMPANY_*` strings instead of `I1MLOG_M101LIB_COMPANY_*` strings.

### **Version 1.28 (June 22, 1999):**

-  Created `TMWdtime.c` and `TMWdtime.h`. Replaced `IECDTIME` type with `TMWDTIME`, a more general date/time structure that is shared with other TMW products. More specifically, the `summertime` member was changed to `dstInEffect`, and the `year` member was changed to a `TMWDEFS_USHORT` that now ranges from 1901 to 2099 instead of 00 to 99.
-  Moved non-IEC-specific date/time functions from the `IECdtime` module to the `TMWdtime` module so that other TMW products can share them. Now takes advantage of `IECdtime` functions such as `iecdtime_store7InMessage()` instead equivalent code inline.
-  Throughout the IEC 60870-5-101 Master Source Code Library, changed the name of functions that included "TimeIEC" to "dateTime," and changed the name of variables that included "IECdTime" to "dateTime."
-  Changed `I1MLOG_FILTER_TRANS` to `I1MLOG_FILTER_TX` in order to be more consistent with other TMW products. Added `I1MLOG_FILTER_RX` to allow filtering only received messages.
-  Changed the order of parameters in `I1Mconfig.h` to be more consistent with configuration modules for other TMW Source Code Libraries.



Removed unnecessary #include's to improve compile time.



Continued revision of source code comments to enhance or more accurately document source code files.



Fixed problem introduced in version 1.26 that prevented freeing of dynamically allocated memory, specifically database memory for a specific sector within a specific session (master to slave link-layer connection). The problem was that the data would have only been freed if the session was on-line, but the sessions were being "reserved" (taken off-line) before calling `i1mdbas_freeSessionData()`. Now, the data for all possible sectors is freed, regardless of the state of the session. Also, `(*i1mdbas_dataSession)[sessionIndex][sectorIndex]` is now initialized to `TMWDEFS_NULL` for all possible sessions and sectors. This ensures that when `i1mdbas_freeSessionData()` is called, `free()` will be called with either `TMWDEFS_NULL`, or a previously allocated memory block.



Fixed additional problem introduced in version 1.26 that prevented calling of `i1mdbas_freeSessionData()` when a session was removed. The problem was that the logic in `i1mdvrs_reserveSession` was reversed.

### **Version 1.27 (June 11, 1999):**



Minor corrections to some source code comments.

### **Version 1.26 (June 9, 1999):**



Added type casts to prevent warnings produced by some compilers.



Changed concept of "device" to "session." A session is a master-to-slave link layer connection. Consequently, changed variable names that use device to use session. Also, `i1mdvrs_session[]` which replaced `i1mdvrs_configDevice[]` is now an array of structures, rather than an array of pointers to structures.



Changed the way "sessions" are installed. No longer are broadcast sessions automatically created. Now all sessions, including broadcast sessions, must be specifically configured by `I1MCONFIG` parameters. Sessions installed during run-time now must use newly created `i1mdvrs_reserveSession()` to obtain a `sessionIndex`, and then `i1mdvrs_installSession()` once the configuration for the `sessionIndex` is setup. Also, `i1mdvrs_reserveSession()` may be used to temporarily take a specific session off-line. Added `I1MDVRS_SESSION_STATE` data type and `sessionState` member to the `i1mdvrs_session[]` array to store the off-line/on-line state of a session.



Created new `I1MCONFIG` parameters: `I1MCONFIG_APPL_PROP_DELAY()`, `I1MCONFIG_DIAG_STRINGS_ENABLED()`, `I1MCONFIG_DIAG_ID_AUX_MASK()`, `I1MCONFIG_PHYS_SESSION_COM_INDEX()`, and `I1MCONFIG_APPL_RESPONSE_TIMEOUT()`.

Changed configuration parameters that were specified using **channelIndex** and **deviceIndex** to being specified with just **sessionIndex**. Changed name of **I1MCONFIG\_PHYS\_NUM\_CHANNELS()** to **I1MCONFIG\_PHYS\_NUM\_COM\_CHANNELS()**.



Changed interface to i1mlog module. Now, all calls to the i1mlog module (except from the i1mbrm module) are made through **I1MLOG\_\*** macros that can be conditionally compiled to be empty depending on the state of **I1MCONFIG\_DIAG\_STRINGS\_ENABLED()**. Removed calls to **i1mlog\_setLogOrigin()**; the information passed to the i1mlog module through this function call is now passed directly through the **I1MLOG\_\*** macros. Also cleaned up i1mlog module, including changing function parameters, and general functional architecture. These changes make it easier to consistently change display formats.



Now uses **tmwprntf\_s()** instead of **sprintf()**. **tmwprntf.c** and other **tmw** modules are now provided as part of the Source Code Library. These modules provide utility support for embedded compilers that may not include such support.



Converted i1mtarg.h module to use macro definitions rather than function prototypes. Macro definitions allow more customization without code-space or execution-time penalties. The **I1MTARG\_\*** macros replaced some **i1mdvrs\_\***() functions and **i1mdbas\_getXX()** and **i1mdbas\_storeXX()** functions.



Enhanced indication of memory allocation errors. Errors detected during initialization are indicated through a new **I1MTRAG\_DIAG\_INIT\_ERROR()** macro and prevent further execution of the source code library. Errors detected during run-time are reported through **I1MLOG\_ERROR()**, and do not stop execution of the library.



For man-machine-interface (i1mbrm.c), changed display of session index and sector index to session link address and sector ASDU address instead.



Changed the type used for **comIndex** in some places from a **TMWDEFS\_UCHAR** to a **TMWDEFS\_USHORT**. This was done for better type consistency, and also to support applications that require more than 256 (either real or virtual) communication ports.

### **Version 1.24 (May 20, 1999):**











Changed name of version stamp variables from **m101\_\*** to **m101lib\_\*** to be more consistent with other TMW products.

### **Version 1.23 (April 6, 1999):**


**101** Added support for negative DEACTCON.






For the MMI defined in i1mbrm.c, added concept of “user commands.” Up to 10 commands (using keys 0 through 9) can be configured to be any supported IEC 60870-5-101 Master ASDU message.

-  The MMI defined in `i1mbrm.c` now initializes the first active device to be a non-broadcast device. Displays were changed to display link and sector addresses (not device indices and sector indices), and to move forward and backward (using the '+' and '-' keys) through configured devices and sectors.
-  Moved initial product and company information display to `i1mbrm` module and now more cleanly centers these displays (using `tmwprntf.c`).
-  Changed `TMW_PREVENT_AUTO_POLL` to `I1MDVRS_PREVENT_AUTO_POLL` defined in `I1Mdvrs.h`.
-  No longer used `i1mdvrs_systemConfig`. Used `I1MCONFIG_*` parameters instead.
-  Added `I1MCONFIG_DBAS_MAX_NUM_DEVICES()` to configure the maximum number of connections with remote devices.
-  Added `I1MLOG_M101_*` literal constants to `i1mlog.h` so that the product name and version can be used by external application-specific software.
-  Fixed problem with allocating database arrays larger than 64K. Changed `i1mdbas_allocSessionData()` to allow `malloc()` to handle allocating more than 64K bytes for compilers that support 32-bit size argument to `malloc()`.
-  Fixed problem in MMI point displays: Now displays last point (use `<=` instead of `<`).


### **Version 1.22 (October 11, 1998):**



-  Under control of `TMW_PREVENT_AUTO_POLL`, added a diagnostic test mode that inhibited continuous class 1/class 2 polls. In this mode, a class 1/class 2 poll can be "forced" with an MMI command. This should be used for initial development only, and no final application code should have it defined.

### **Version 1.20 (June 8, 1998):**



-  Changed some default configuration values to allow easier testing with the Triangle MicroWorks 101 Slave Source Code Library.
-  Fixed problem with information object address size of 3.
-  Removed `class1Counter` from `I1MLINK_DEVICE_MGR.rec` because it was only reset when a new request sent or no class 2 data available. This is a problem if the remote device always has class 2 cyclic data available.

### **Version 1.18 (April 24, 1998):**



-  Renamed all files and all public symbols from `ONExxxx.*` to `I1Mxxxx.*`.

-  Removed Microsoft Windows specific function calls from i1mdvrs.c and replaced them with example target-specific calls, as prototyped in i1mtarg.h.
-  Now includes makefile as an example of project dependencies.





### **Version 1.16 (March 22, 1998):**




-  In `onetask_queHandler()`, moved processing of data link confirm timeouts in front of application layer transmit queue processing so timeout on link confirm gives higher priority to application layer frame than link class1/2 poll frame.
-  Add function `onedvrs_getMsTime()` instead of using `onedvrs_getSystemTime()` to read free running millisecond counter so that timeout counters are not affected by changing time on target platform. Remove function `iecdtime_msecDifference()`.

### **Version 1.14 (September 11, 1997):**

- 101 When requesting user data class 1 or 2, interpret single character ACK response as NACK: requested data not available.
-  Changed link transmit variable reset (true/false) to linkState: `LINK_CONNECTION_INIT`, `LINK_CONNECTION_FAILED`, or `LINK_CONNECTION_ESTABLISHED`.
-  Fix notification of link connection failure after new device is installed.

### **Version 1.12 (August 12, 1997):**


- 101 Added support for bitstring data types.
- 101 Added support for delay measurement and time synchronization.
- 101 Added support for broadcast messages.
- 101 Added support for single character ACK.
-  Renamed TMWdtime module to IECdtime and updated symbol names defined within that module. Removed `iecdtime_computeDayOfWeek()` since it is never called. Changed `dayOfWeek` meaning to 0 = not used, 1 = Monday. Added function `iecdtime_msecDifference()`.
-  Added Motorola (most-significant-byte-first) versions of `getxx()` and `storexx()` data conversion functions.
-  Now calls `onebrm_endofInitcallback()` upon receiving end-of-initialization message.
-  Added test for ASDU field size configuration errors in `onedca_readDataUnitId()`.

-  Enhanced onelog module by changing `onelog_time2string()` to use `onedvrs_getsystemTime()`. Added `onelog_showSingleLinkRec()`. Added `onedvrs_errorHandler()` to notify target application as error conditions occur.
-  Added calls to `onedvrs_errorHandler()` when link connection fails or is (re)established.
-  Eliminated `onedvrs_syncTimers()` by calling `onedvrs_getsystemTime()` from `onetask_libraryTimers()` and calculating elapsed time directly from `mSecsAndSecs`.

### **Version 1.10 (June 11, 1997):**

-  Replaced function `onedvrs_storeCurrentTime()` with `onedvrs_getsystemTime()` and new link module function `storeTimeInMessage()`.

### **Version 1.08 (May 28, 1997):**

-  Initial release of the IEC 60870-5-101 Master Source Code Library.