

EXTENSION TOOLKIT FOR DASYLAB 2016 AND 2022

INHALT (CONTENT)

Liste der Änderungen (<i>List of changes</i>).....	3
Toolkit Version: 1 -> 2 for DASYLAB 2016 and 2 -> 3 for DASYLAB 2022	3
DLAB_Float: FLOAT -> Double (only DASYLab 2016 – no Further changes for 2022)	3
Max. BlockSize (only DASYLab 2016 – no Further changes for 2022)	4
Sysinfo -> Serialopt (only DASYLab 2016 – no Further changes for 2022)	4
Entfernte Veraltete Funktionen (<i>Removed deprecated Functions</i>) - (only DASYLab 2016 – no Further changes for 2022).....	4
Geänderte Strukturen / Funktionen (<i>Slightly changed Structures / Functions</i>) (only DASYLab 2016 – no Further changes for 2022).....	5
Vorbereitungen für spätere DASYLab Versionen (<i>Compatibility Preparation for later DASYLab versions</i>) – Removed for DASYLab 2022 – Sorry for Inconvenience.....	5
Neue Modul Message für Datenquellen (New Module Message for Data Source Modules – including DLL driver modules) (only DASYLab 2016 – no Further changes for 2022)	6
Neue Modul Message für ALLE MODULE – benutzt vom SMART MULTIPLEXING (<i>New Module Message for ALL Modules</i>) (only DASYLab 2022).....	6
Hinweise für die Umstellung (<i>Hints for Modification of your DASYLab DLL</i>)	7

Für die DASYLab Version 2016 (DASYLab 14) mussten einige gravierende Änderungen an der Schnittstelle zwischen DASYLab und den Modulen durchgeführt werden. Diese Änderungen führen dazu, dass ältere DASYLab Erweiterungen nicht mehr mit DASYLab 2016 kompatibel sind. Sie können bewusst **nicht** in DASYLab 2016 geladen werden. Ebenso sind Module, die mit dieser Version des Toolkits erstellt wurden, nicht mit älteren DASYLab Versionen kompatibel (DASYLab 13 oder früher). Dieses die originale Dokumentation ergänzende Dokument beschreibt die Änderungen an der Schnittstelle zu DASYLab und gibt Hinweise zu den notwendigen durchzuführenden Änderungen.

Für die DASYLab Version 2022 (DASYLab 16) wurden ehemals geplante – aber durch andere Ansätze überholte – Funktionsdeklaration und deren Import aus dem Hauptprogramm entfernt / geändert / erweitert. Dies bedingt eine Neucompilierung mit dem DASYLab 2022 Toolkit und wenige einfache Änderungen (kleiner 2 Minuten pro Modul).

*For DASYLab Release 2016 (DASYLab 14) the interface between the main program and the DLLs with the modules went through significant modifications/enhancements. As a result DLLs created with the extension toolkit in an earlier release (< 14) are not compatible with DASYLab 14. They will not load **intentionally** – vice versa DLLs created with Extension Toolkit 14 will not load in DASYLab 13 and earlier. This document lists and describes the modifications of the interface and is an addendum to the original documentation. Look for hints for the necessary steps.*

For DASYLab Release 2022 (DASYLab 16) the interface between the main program and the DLLs with the modules went through slightly reductions/modifications/enhancements. As a result, DLLs created with the extension toolkit in an earlier release (< 16) are not compatible with DASYLab 2022 and need to recompile with the recent toolkit. Only a few simple changes are necessary (less than 2 minutes per module).

LISTE DER ÄNDERUNGEN (LIST OF CHANGES)

Es folgt nun eine Liste mit den Änderungen am Toolkit und einer Erklärung dazu. Im Folgenden gehen wir auf die zentralen Änderungen genauer ein und geben ihnen Hinweise, was Sie bei der Umsetzung ihrer DASYLab Erweiterung auf die neue Version beachten müssen.

In the next small sections we list the changes in the toolkit and provide hints, what actions are needed to migrate your source code to the next version of the toolkit and DASYLab.

TOOLKIT VERSION: 1 -> 2 FOR DASYLAB 2016 AND 2 -> 3 FOR DASYLAB 2022

Durch die Erhöhung der internen Toolkit Version wird sichergestellt, dass ältere DASYLab Versionen keine mit diesem Toolkit erstellten DLLs laden können. Ebenso erwartet DASYLab 2016 bzw. DASYLab 2022 jetzt auch DLLs, die mit Toolkit Version 2 bzw. 3 erstellt wurden und lädt keine älteren DLLs mehr. Die Beispiel Module wurden angepasst.

Nötige Aktionen bei der Migration: keine

By changing the internal Toolkit version (i.e. the supported toolkit version in DASYLab) we make sure, that DASYLab 14 or 2022 can only load DLLs created with toolkit version 2 or 3 and – vice versa – older DASYLab releases can only load DLLs created with older releases of the toolkit. This is intentionally because a lot of the data structures changed – most of them transparent to you as the developer of the DLL. The example modules reflect the necessary changes.

Necessary actions for migration: none

DLAB_FLOAT: FLOAT -> DOUBLE (ONLY DASYLAB 2016 – NO FURTHER CHANGES FOR 2022)

Der Datentyp für DLAB_FLOAT wurde von float zu double geändert. Durch diese Änderung werden alle Daten nun mit doppelter Genauigkeit (8 Byte Fließkommazahlen gegenüber 4 Byte Fließkommazahlen) verwaltet und zwischen den Modulen ausgetauscht – dies erhöht den Speicherbedarf eines Schaltbildes zur Laufzeit grob um den Faktor 2. Diese Änderung bringt auch Änderungen an weiteren, zentralen Datenstrukturen (z.B. DATA_BLOCK_HEADER) mit sich und ist der Hauptgrund für die Inkompatibilität von DLLs, die mit diesem Toolkit erstellt wurden, zu älteren DASYLab Versionen und umgekehrt.

Durch die erhöhte Genauigkeit werden Messdaten und Berechnungen nicht zwischen den Modulen auf 6-7 stellige Genauigkeit abgeschnitten – die Kumulierung arithmetischer Fehler wird vermindert und Arbeiten mit Zählerwerten wird verbessert.

Nötige Aktionen bei der Migration: Alle Typkonvertierungen zu float überprüfen (unnötige Typkonvertierungen entfernen) – nach allen Vorkommen des Schlüsselwortes „float“ suchen und ggf. korrigierend eingreifen.

DLAB_FLOAT was changed from 4 byte single to 8 byte double to improve the accuracy of computation results and minimize errors generated by accumulation of computing errors. On the other hand this modification will roughly double the necessary amount of memory for DASYLab flowcharts at runtime for DASYLab 2016 (there are plans for DASYLab “15” to take care of this possible issue in optimization of the internal memory management, if this is a “real” and not a theoretic issue). Beside better accuracy the user gains a better handling of any counted value (hardware ore internal counter) – in 32bit DLAB_FLOAT any integer will start to loose digits (counts every 2nd, 3rd 4th, ... value) after a while. This happens now much later (6-7 digits of precision vs 15-16 digits of precision).

Necessary actions for migration:

Delete all explicit type conversions, which are not really necessary.
Search for all occurrences of "float" and improve, if necessary.

MAX. BLOCKSIZE (ONLY DASYLAB 2016 – NO FURTHER CHANGES FOR 2022)

In der Struktur „DATA_BLOCK_HEADER“ wurde das Element „wBlockSize“ in „uiBlockSize“ geändert. Der Typ ist nun UINT, statt WORD. Ein Datenblock kann nun also größer als 32768 Samples sein.

In der Struktur „FIFO_HEADER“ wurde korrespondierend zu dem „DATA_BLOCK_HEADER“ das Element „wBlockSize“ in „uiBlockSize“ umbenannt. Der Typ war schon vorher „UINT“.

Nötige Aktionen bei der Migration:

Führen Sie die notwendigen Umbenennungen durch stellen Sie sicher, dass Ihre Module auch mit Blockgrößen größer 32768 korrekt funktionieren. Die Maximalgröße in DASYLab 2016 wird $2 \text{ hoch } 20$ (1 MSample) sein. Zum Testen kann das Datenfenster Blöcke verschiedener Größen erzeugen.

In „DATA_BLOCK_HEADER“ the member „wBlockSize“ was renamed to „uiBlockSize“ and its data format was enlarged from WORD to UINT. Block sizes can be larger than 32768 Samples now.

In „FIFO_HEADER“ the member „wBlockSize“ was renamed to „uiBlockSize“ – data format was already „UINT“.

Necessary actions for migration:

Rename the variables in your code and make sure your modules work with block sizes larger than 32768. The maximum block size for DASYLab 2016 is $2 \text{ in the power of } 20$ (1 MSample). For test purposes use the data window – this is included in your test version and can produce block sizes larger than 32768.

SYSINFO -> SERIALOPT (ONLY DASYLAB 2016 – NO FURTHER CHANGES FOR 2022)

Die Struktur „Sysinfo“ heißt nun „SerialOpt“ und beinhaltet mehr Informationen im Klartext als Optionen. Entfallene Optionen sind entsprechend umbenannt und Reserve für zukünftige Erweiterungen.

Nötige Aktionen bei der Migration: keine

The structure „Sysinfo“ was responsible for authentication of DLLs regarding flags in the serial number. Since most of the work is done in the main program at startup time, you probably don't need to care. However - it is renamed to „SerialOpt“ and gained some more flags.

Necessary actions for migration:

none

ENTFERNTE VERALTETE FUNKTIONEN (REMOVED DEPRECATED FUNCTIONS) - (ONLY DASYLAB 2016 – NO FURTHER CHANGES FOR 2022)

Die Funktionen "ShowPipeStatus" und "AdvancePipeStatus" wurden entfernt. DASYLab behandelt die Datenfluss Animation schon seit längerem intern. Die Funktionen hatten keinen Effekt. Die Funktion „SetFontSmallProc“ wurde entfernt, da sie noch aus der 16 Bit Zeit stammte. Die Funktion „CenterDialog“ wurde entfernt. Diese war schon seit längerem nur noch ein funktionsloser Rumpf ohne Auswirkung. Funktionsdeklarationen zum Laden und Speichern von Bitmaps in verschiedenen Formaten (die allerdings nie von DASYLab exportiert wurden), wurden entfernt

Nötige Aktionen bei der Migration:

Löschen der überflüssigen Aufrufe.

"ShowPipeStatus" and "AdvancePipeStatus" were removed, since DASYLab handles the animation of the data flow internally. „SetFontSmallProc“ was removed, since deprecated from Windows 32 on. „CenterDialog“ was

removed, since handled in DASYLab internally. Some function declarations (that were never exported to this interface) for bitmap handling were removed.

Necessary actions for migration: Delete function calls in your code

GEÄNDERTE STRUKTUREN / FUNKTIONEN (SLIGHTLY CHANGED STRUCTURES / FUNCTIONS) (ONLY DASYLAB 2016 – NO FURTHER CHANGES FOR 2022)

- Die Strukturen „MODUL“, „MODULKLASSE“ und „FENSTERPARAMETER“ wurde erweitert.
- Es wurden neue Konstanten für „ACTION_PARAM_TYPE_xxx“ hinzugefügt.
- Konstanten für die vergebenen Zeitbasen IDs wurden ergänzt.
- Die Funktion „ModalChooseFontDisableSize“ wurde geändert. Es gibt nun keine Hook Funktion mehr.
- Die Struktur „ToolTipTyp“ wurde erweitert. Der String hat nun eine maximale Länge von 512 Zeichen.
- Die Funktion „CheckWindowsPosition“ wird nun von DASYLab bereitgestellt und hat einen Parameter mehr.
- Es wurde eine neue globale Variable „chDefSharedUserDir“ hinzugefügt. Diese enthält den Pfad auf das Basis Verzeichnis für variable DASYLab Dokumente. Hier ist z.B. seit DASYLab 10 auch die DASYLab INI Datei gespeichert.

Nötige Aktionen bei der Migration: keine – aber teilweise Neuverwendung möglich

- Internal structures „MODUL“, „MODULKLASSE“ and „FENSTERPARAMETER“ enhanced.
- New constants for event actions „ACTION_PARAM_TYPE_xxx“.
- New constants for time base IDs (need to be mutually exclusive for all time sources in the system)
- Removed hook function (Windows 16 bit remainder) in „ModalChooseFontDisableSize“.
- „ToolTipTyp“ has a string length of 512 chars now.
- „CheckWindowsPosition“ is exported by DASYLab now.
- New global variable „chDefSharedUserDir“ that provides the path tot he base directory for DASYLab document files. In this location the DASYLab.INI is located (since DASYLab 10).

Necessary actions for migration: none – but use, if applicable

VORBEREITUNGEN FÜR SPÄTERE DASYLAB VERSIONEN (COMPATIBILITY PREPARATION FOR LATER DASYLAB VERSIONS) – REMOVED FOR DASYLAB 2022 – SORRY FOR INCONVENIENCE

Für spätere DASYLab Versionen (ab DASYLab „15“) wird ein neues API zum Durchschleusen von statischen und dynamischen Informationen – zusätzlich zu den Daten und Aktionen – entwickelt. Eine erste Rohversion des API ist zur Ansicht enthalten – DASYLab 2016 Module können jetzt schon eine Funktion benutzen, um die spätere Funktionalität in DASYLab passiv zu unterstützen. Diese Funktion sollte vor dem Aufruf von ReleaseOutputBlock durchgeführt werden. Anbei ein Ausschnitt aus dem Source Code des Toolkits (deriv.c).

For later DASYLab releases (DASYLab 15 and later) we provide a new API for data transport at start time (we call this static) and data transport with the data blocks at process time (we call that dynamic). A first version of this API is included in the toolkit, but all functions (except one) are subject to change. To provide compatibility with later DASYLab releases (as far as we can look into the future) add a call of EmemBlock_PROCESS_MsgCopyPlain before ReleaseOutputBlock. Beneath you'll find example code (taken from the deriv.c) of the toolkit.

```
// TK14: For future compatibility: Copy extra memory (per data block) from the "Father" block to the block we filled with data above
//      : This will have no effect in DASYLab 14 (2016) but your module/dll will support a new feature in DASYLab 15 without recompiling
//      : The cpu cost for DASYLab 14 is near zero – the call of this funtion is optional for DASYLab 14 and mandatory for DASYLab 15

// If the InFifo parameter is NULL, the function assumes, that the maximum blocksize of the input fifo is equal to the maximum blocksize
// of the output fifo (what is "the normal behaviour"). If you are unsure (what you shouldn't be because you wrote/modified the
// SetupFifo_xxx routine) or the maximum blocksize of OutFifo and InFifo is not equal, the provide the InFifo as parameter.
EmemBlock_PROCESS_MsgCopyPlain (OutFifo, OutputBlock, NULL, InputBlock);

// Add this Data Block to the FIFO, so that a "Son" FIFO can get Access to it
ReleaseOutputBlock (OutFifo);
```

NEUE MODUL MESSAGE FÜR DATENQUELLEN (NEW MODULE MESSAGE FOR DATA SOURCE MODULES – INCLUDING DLL DRIVER MODULES) (ONLY DASYPOL 2016 – NO FURTHER CHANGES FOR 2022)

Um nur die benutzten Zeitbasen im Schaltbild abzuspeichern, fragt DASYPOL beim Abspeichern alle Module, ob (und welche) Zeitbasis sie verwenden. Die Default Funktion (wenn Sie nichts tun) liefert dabei 0 als Wert zurück (keine Zeitbasis wird verwendet) – die datenerzeugenden Module sollten die ID der verwendeten Zeitbasis zurückgeben. Falls das nicht gemacht wird, werden Abtastrate und Blockgröße dieser Zeitbasis nicht im Schaltbild abgespeichert. Dies wurde in der Datei „Generat.c“ des Toolkits Beispiels entsprechend realisiert.

Anmerkung: die Zeitbasen DASYPOL und Treiber werden immer abgespeichert – also kein Handlungsbedarf für veraltete Treiber Toolkit Treiber.

Nötige Aktionen bei der Migration: Hinzufügen der Behandlung der Meldung `DMM_GET_TIMEBASE_ID`.

DASYPOL 2016 stores only the used time bases into the worksheet per default (you can change that to store all time sources in the system as it was before in the global settings dialog box). For this purpose it needs to ask every module, which time base ID it uses. The default function returns 0 (so no time base is used) what is appropriate for all data processing modules. Nothing to do for you.

But all data source modules (slider, switch, generator, hardware input modules, hardware output modules) need to return the time base ID, they are bind to (or they are synchronized to). If you do not answer to that message, the block size and the sample rate of your time base do not make it into the work sheet. Example is in Generat.c in this toolkit.

Hint: The time bases DASYPOL and Driver are always stored into the flowchart – so nothing to do for old fashioned Driver toolkit drivers.

Necessary actions for migration: Add handling of message `DMM_GET_TIMEBASE_ID`

Beispiel aus Generat.c / Example from Generat.c

```
case DMM_GET_TIMEBASE_ID:  
    return ((MODULE_GENERATOR *)ThisModule->ModuleParameters)->uiTimeBase;
```

NEUE MODUL MESSAGE FÜR ALLE MODULE – BENUTZT VOM SMART MULTIPLEXING (NEW MODULE MESSAGE FOR ALL MODULES) (ONLY DASYPOL 2022)

In DASYPOL werden (vereinfacht gesagt) die Strukturinformationen der einzelnen Module beim Strukturtest von links nach rechts durch das Schaltbild transportiert und gemäß der Funktion der Module angepasst (wie später der Datenstrom auch). Für eine performante und effektive Speicherverwendung beim Smart-Multiplexing, benötigen einige DASYPOL Funktionen aber auch „einen Blick“ auf das Quellmodul am Eingang, um dort Kanalnamen und Einheiten zu erfragen. Dazu ist die Modul Message (genauer Kanal Message) `DCM_GET_CHANNEL_NAME` zu verwenden.

Anmerkung: Falls Sie sich nun wundern, dass es eine sehr ähnlich klingende Modulmessage schon gibt – die „alte“ Modul Message `DMM_GET_CHANNEL_NAME` hätte vor langer Zeit einen besseren Namen verdient – `DCM_COPY_CHANNEL_NAME_ON_USER_FLAG` trifft es besser. Sie können diese (numerisch identische) Konstante ersetzen – oder eben auch nicht. In den Beispielen ist das der Klarheit wegen angepasst.

Nötige Aktionen bei der Migration: Hinzufügen der Behandlung der Meldung `DCM_GET_CHANNEL_NAME`.

Similar to data flow (while processing), DASyLab carries the structure information of the modules from left to right through the flowchart (with adjustments – depending on the function of the module). The smart multiplexing feature works different – it does not copy the information about channel name and unit directly to the receiver; instead, it provides a link to the information (internally a tree) and needs to investigate the “left” module while dissolving that link. Therefore, it sends a `DCM_GET_CHANNEL_NAME` message to that module.

Remark: When you’re amazed about the very similar `DMM_GET_CHANNEL_NAME` message – a long time ago we could have given that a better name – `DCM_COPY_CHANNEL_NAME_ON_USER_FLAG` would represent the functionality better. Feel free to replace `DMM_GET_CHANNEL_NAME` with `DCM_COPY_CHANNEL_NAME_ON_USER_FLAG` or leave it as it is. We changed the examples for clarity, but it has no other effect.

Necessary actions for migration: Add handling of message `DMM_GET_TIMEBASE_ID`

Beispiel aus `Generat.c` / Example from `Generat.c`

```
case DCM_GET_CHANNEL_NAME:
    return (long)((MODULE_GENERATOR*)ThisModule->ModuleParameters)->szChannelName[wParam];
```

HINWEISE FÜR DIE UMSTELLUNG (HINTS FOR MODIFICATION OF YOUR DASyLAB DLL)

- Lassen Sie Visual C so viel Arbeit machen wie möglich – benutzen Sie Warnstufe 4
- Vermeiden Sie explizite Typkonvertierungen (wenn möglich) – wenn der Compiler ohne nicht weiß was Sie wollen, könnte es eine Problemstelle sein.
- Für Fragen oder Hilfe oder Anregungen nutzen Sie bitte das DASyLab Forum – da gibt es einen Bereich für die Arbeit mit dem Toolkit.
<http://www.dasylab-forum.de/forum/index.php>

- Push as much work as possible to the development environment – use the maximum warning level
- Prevent unnecessary explicit type conversion – if the compiler doesn’t know without, it might be a problem.
- For questions or help or suggestions use the DASyLab forum. There is a sub forum called Toolkit.
<http://www.dasylab-forum.de/forum/index.php>