
AS Data Manager

Anwenderhandbuch

Titel	AS Data Manager
Dokumenttyp	Anwenderhandbuch
Dateireferenz	AS_Datamanager.library
Version	2.1
Datum	22.06.2026
Autor	ASKS GmbH
Status	Freigegeben

ASKS GmbH · Tevesstraße 29a · 78176 Blumberg · www.as-ks.de · support@as-ks.de

© 2026 ASKS GmbH. Alle Rechte vorbehalten.

Inhaltsverzeichnis

Inhalt

Inhaltsverzeichnis.....	2
1 Übersicht.....	3
2 Inbetriebnahme.....	3
2.1 Applikation	3
2.2 Datamanager initialisieren	5
2.3 Gruppen	5
2.4 Werte	6
2.5 Speichern und Laden	7
2.6 Übersetzung.....	7
3 Visualisierung	8
3.1 Vorbereitungen	8
3.2 Datensatz	9
3.3 Werte	10
3.4 Übersetzung	11
3.4.1 Beispiel Teiltextauflösung	11
3.5 Einheitenkonvertierung	12
3.5.1 Beispiel Einheitenkonvertierung	12
4 Lizenz	13

Zusammenfassung: Dieses Anwenderhandbuch beschreibt den ASKS Data Manager – eine CODESYS-Bibliothek zur persistenten Speicherung und Wiederherstellung von SPS-Parametern (Rezepte, Formate, Maschinenparameter). Es erklärt Einbindung, Initialisierung, Datenstrukturierung, Speicher- und Ladevorgänge sowie die bereitgestellten Visualisierungen und Erweiterungsmöglichkeiten.

1 Übersicht

Der AS Data Manager ermöglicht die persistente Speicherung und Wiederherstellung von Parametern in einer SPS – beispielsweise Rezepte, Formate oder Maschinenparameter. Die Daten werden standardmäßig in einer CSV-Datei gespeichert. Weitere Speicherformate können auf Anfrage implementiert werden.

Die wichtigsten Funktionen der Bibliothek umfassen das Speichern und Wiederherstellen von Datensätzen in CSV-Dateien, eine automatische Datensicherung beim Speichern sowie die Möglichkeit, benutzerdefinierte Datentypen zu speichern und wiederherzustellen. Gruppen und Werte lassen sich beliebig verschachteln. Änderungen im Datamanager, in Gruppen oder Einzelwerten werden per Benachrichtigung gemeldet. Für Datamanager, Gruppen, Gruppenlisten, Wertelisten und Einzelwerte stehen separate Visualisierungen bereit. Die Bibliothek unterstützt eine clientspezifische Übersetzung sowie eine erweiterbare Einheitenkonvertierung.

Alle Fehlercodes der Bibliothek sind in der ERROR-Struktur definiert. Die Bibliothek AS_Datamanager stellt die grundlegenden Datentypen bereit.

2 Inbetriebnahme

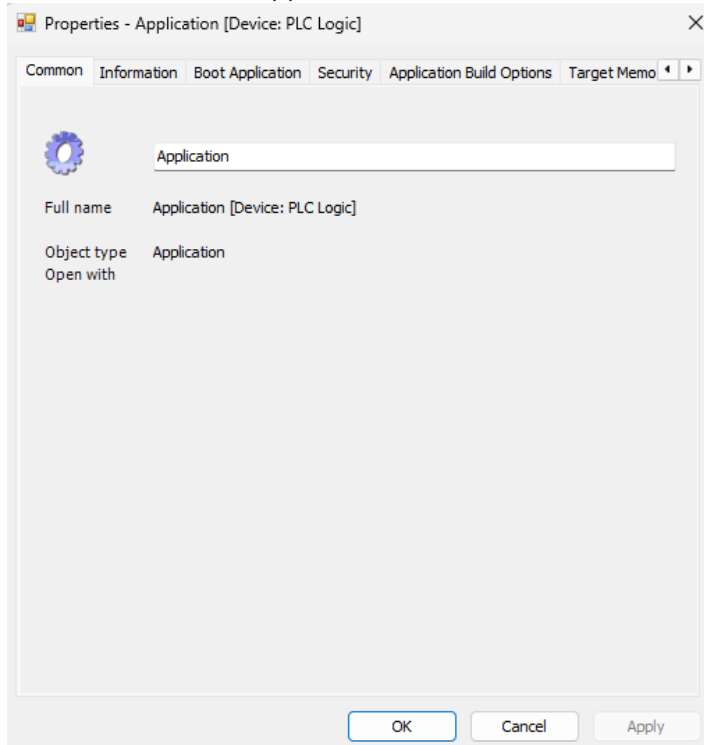
In diesem Kapitel wird beschrieben, wie der Data Manager in ein CODESYS-Projekt eingebunden und konfiguriert wird.

2.1 Applikation

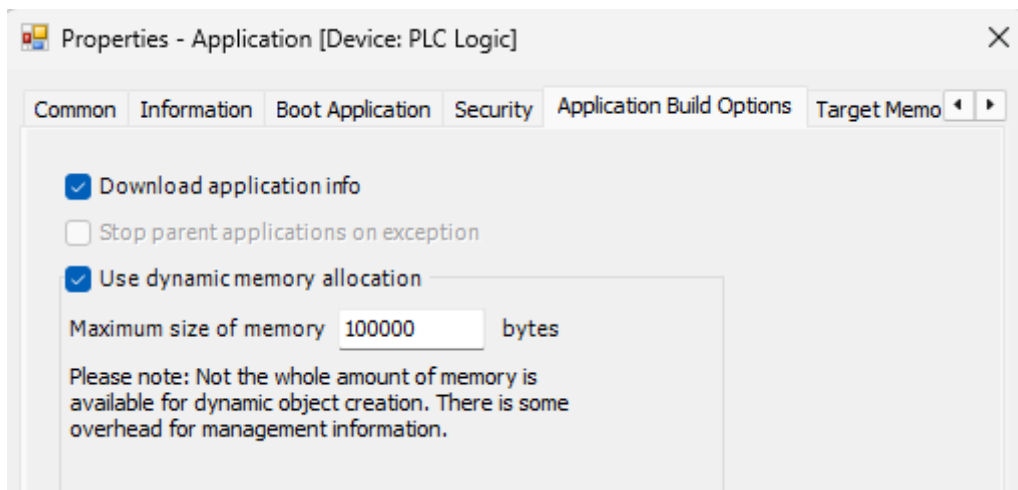
Damit die Datamanager-Funktionen genutzt werden können, muss die Bibliothek AS_Datamanager zunächst im Library Manager eingebunden werden.

	3SLicense = 3SLicense, 3.5.20.0 (CODESYS)	_3S_LICENSE	3.5.20.0
	AS_Datamanager = AS_Datamanager, 0.0.0.0 (ASKS GmbH)	DAM	0.0.0.0 

Anschließend muss der dynamische Speicher der Applikation vergrößert werden. Dazu wird unter dem Knoten „Application“ über das Kontextmenü die Option „Properties“ aufgerufen.



Unter „Application Build Options“ ist die Option „Use dynamic memory allocation“ zu aktivieren und die maximale Größe zu erhöhen.



HINWEIS: Ohne die Erweiterung des dynamischen Speichers kann der Datamanager nicht initialisiert werden

2.2 Datamanager initialisieren

Eine Datamanager-Instanz wird im CODESYS-Programm deklariert und anschließend initialisiert. Die Instanz verwaltet alle Gruppen und Werte sowie die Speicherzugriffe.

```
Recipes.Init(                                     // File - Recipe Management
    sName      := 'Recipe Management',           // Datamanager Name
    sTranslationFile := 'txtTestTranslations',    // [] Translation file name
    sDir        := 'Recipes',                     // This is the directory where to save/load the files
    sFileType    := 'csv',                       // [dmf] This is the name of the ending (txt, csv, ...)
    xDoBackup     := TRUE,                       // [FALSE] Datamanager is doing a backup if there are pending changes on save
    xLoadMinMax   := FALSE;                     // [FALSE] Load the Min - Max limits form the file
```

2.3 Gruppen

Gruppen strukturieren die Datensätze hierarchisch. Sie können beliebig tief verschachtelt werden (siehe Hinweis) und ermöglichen so eine klare Gliederung komplexer Parametersätze. Eine Gruppe wird über ihren Konstruktor initialisiert und dem Datamanager oder einer übergeordneten Gruppe hinzugefügt.

```
Granulation.InitGroup(                          // Group - Granulation
    sName      := 'GRANULATION MIX',             // Group Name
    sTranslationFile := 'txtTestTranslations',    // [] Translation file name
    parent      := Linel);                      // Parent Group

Steril.InitGroup(                               // Group - Sterilization
    sName      := 'STERILIZATION CYCLE',         // Group Name
    sTranslationFile := 'txtTestTranslations',    // [] Translation file name
    parent      := Linel);                      // Parent Group
```

HINWEIS: Die maximale Pfad länge darf 255 Zeichen nicht überschreiten.

2.4 Werte

Der Datamanager unterstützt die folgenden Datentypen: BOOL, DINT, ENUM, INT, REAL, SINT, STRING, UDINT, UINT, USINT und TIME. Einige Initialisierungsparameter sind optional; Standardwerte sind in der Referenzdokumentation angegeben. Hier einige Beispiele:

```
ExampleBool.Init(
    group      := ExampleGrp,           // Parent Group
    sName      := 'EXAMPLE BOOL',      // Value Name
    xDefault   := FALSE,               // Default Value
    sUnit      := '');                // [''] Unit of the Value

ExampleInt.Init(
    group      := ExampleGrp,           // Parent Group
    sName      := 'EXAMPLE INT',       // Value Name
    nMin       := 0,                   // [1] Min Value
    nDefault   := 10,                  // Default Value
    nMax       := 100,                 // [0] Max Value
    sUnit      := '');                // [''] Unit of the Value

ExampleReal.Init(
    group      := ExampleGrp,           // Parent Group
    sName      := 'EXAMPLE REAL',      // Value Name
    fMin       := 0.0,                 // [1] Min Value
    fDefault   := 10.0,                 // Default Value
    fMax       := 100.0,                // [0] Max Value
    sUnit      := '',                  // [''] Unit of the Value
    nDecimals  := 2);                 // [3] Decimal places
```

*Initialisierung einiger Data-Manager-Werte (optionale Parameter in eckigen Klammern)
Alle Values werden im Beispiel Projekt aufgelistet und initialisiert.*

Über Eigenschaften (Properties) des jeweiligen Funktionsbausteins kann zur Laufzeit auf den aktuellen Wert zugegriffen werden.

```
x := ExampleBool.xValue;
i := ExampleInt.nValue;
f := ExampleReal.fValue;
n := ExampleEnum.nValue;
str := ExampleString.sValue;
```

2.5 Speichern und Laden

Um einen Datensatz zu speichern oder zu laden, werden die Methoden `Save()` bzw. `Load()` aufgerufen und der gewünschte Datensatzname übergeben. Fehlende Werte beim Laden werden ignoriert oder können – je nach Konfiguration – auf den Standardwert gesetzt werden.

```
IF xSave THEN
    xSave := FALSE;
    Recipes.Save('Recipes');
END_IF

IF xLoad THEN
    xLoad := FALSE;
    Recipes.Load('Recipes');
END_IF
```

2.6 Übersetzung

Die Übersetzung kann auf zwei Arten erfolgen: über den globalen Sprachhandler (anwendungsweit) oder über eine gruppenspezifische Textliste. Beide Ansätze können innerhalb derselben Anwendung kombiniert werden.

Globale Übersetzung

Eine Sprachdatei wird einmal über `DAM.LAN.GV_LAN.g_language.AddGlobalLanguageFile()` registriert und steht anschließend allen Gruppen zur Verfügung. Dies geschieht typischerweise während der Initialisierung der Anwendung.

```
DAM.LAN.GV_LAN.g_language.AddGlobalLanguageFile('txtTestTranslations'); //Global Language File for Translations
```

Gruppenspezifische Übersetzung

Einer Gruppe kann über den Parameter `sTranslationFile` in `InitGroup()` eine eigene Textliste zugewiesen werden. Diese überschreibt die globale Datei für Zeichenketten im Gültigkeitsbereich dieser Gruppe und ermöglicht so eine modulare, in sich geschlossene Übersetzung pro Gruppe.

```
ExampleGrp.InitGroup(
    sName          := 'EXAMPLES',
    sTranslationFile:= 'txtExample',
    parent         := grp);
```

HINWEIS: Namenräume : DAM = AS Data Manager, LAN = AS Language

3 Visualisierung

Die in diesem Kapitel beschriebenen Visualisierungen sind Open-Source-Vorlagen, die unverändert übernommen oder nach Kundenwunsch angepasst werden können.

3.1 Vorbereitungen

Für die Nutzung der DM-Visualisierung sind drei Konfigurationsschritte erforderlich.

Im Visualisierungsmanager ist die Option „Download all Dialogs“ zu aktivieren. Das geschieht durch das Anhängen der benötigten Dialoge.

Name	WebVisu	Number of Instances
Default Behavior	☐	
Visualizations		
Datananager	☒	
Dialogs		
DAM		
VIS		
Binpad	☒	1
Enumpad	☒	1
Keypad	☒	1
Numpad	☒	1
Timepad	☒	1
VisUserManagement		
VUM_ChangePassword	☐	1
VUM_Login	☐	1
VUM_LoginVisualization	☐	1
VUM_UserManagement	☐	1
VUM_UserManagementWide	☐	1
VisDialogs		
FileOpenSave	☐	1
Keypad	☒	1
Login	☐	1
MessageBox	☐	1
Numpad	☒	1
NumpadExtended	☐	1
TextinputWithLimits	☒	1

In der VISU_TASK muss die Methode updateVisu() des ASKS Visualization Handlers aufgerufen werden.

Configuration

Priority (0..31): Task group

Type Interval (e.g. t#200ms)

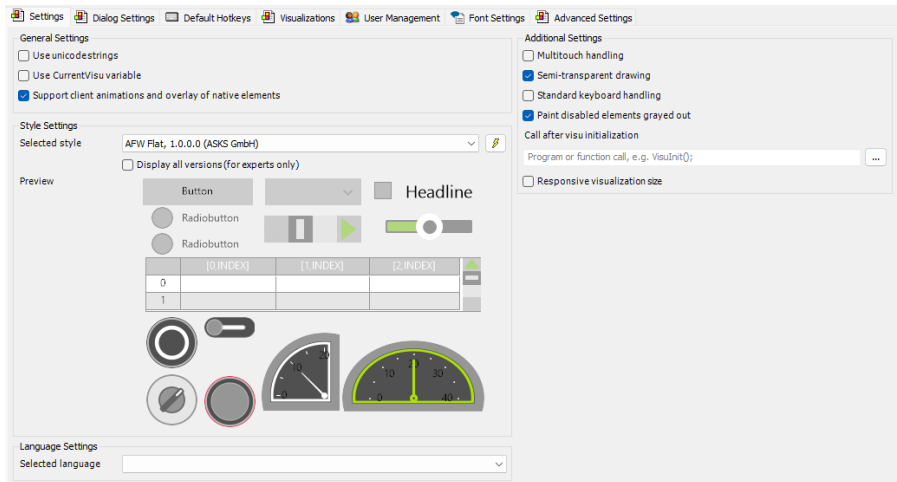
Watchdog ☐ Enable

Time (e.g. t#200ms)

Sensitivity

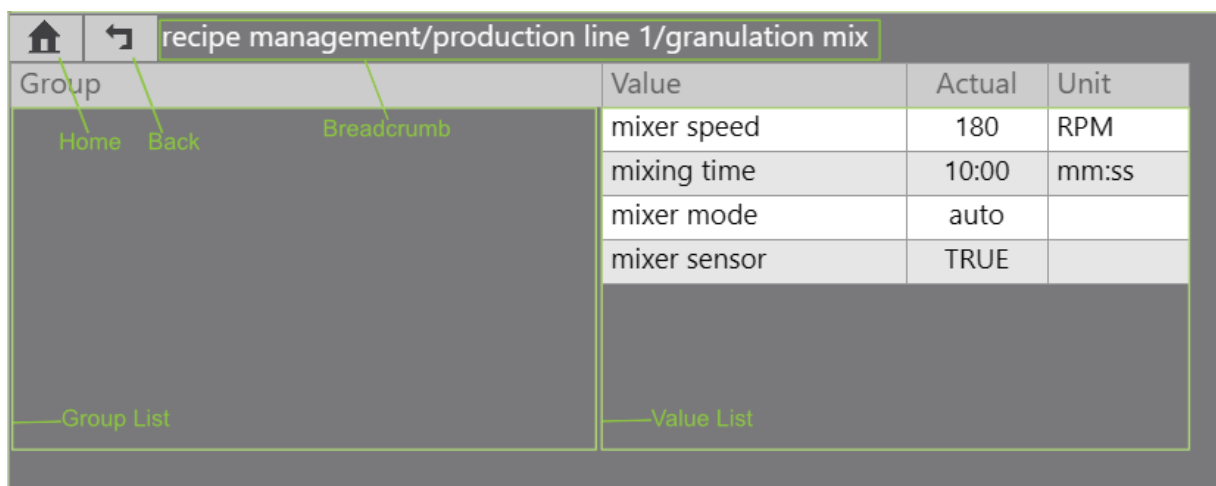
POU	Comment
 VisuElems.Visu_Prg	
 DAM.VIS.VisuHandler.UpdateVisu	

Als Visualisierungsstil ist „AFW Flat“ im Visualisierungsmanager zu wählen. Anpassungen an den Visualisierungen können auf Anfrage bei ASKS bezogen werden.

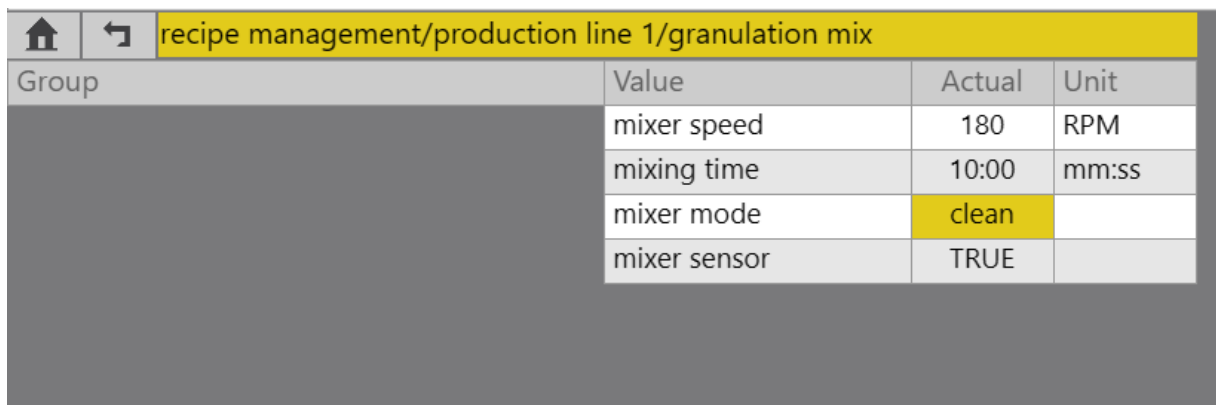


3.2 Datensatz

Um einen vollständigen Datensatz anzuzeigen, wird die Visualisierung VisuDm in einem Frame instanziiert und die gewünschte Gruppe bzw. der Datamanager übergeben. Gruppen und Untergruppen ohne enthaltene Werte werden automatisch ausgeblendet.



Eine gelbe Färbung zeigt an, dass nicht gespeicherte Änderungen im Datensatz vorhanden sind.



3.3 Werte

Zusätzlich zu den Standard-CODESYS-Eingabedialogen stellt der DM eigene Dialoge bereit, die die Eingabe spezifischer Werte vereinfachen.

mixer sensor

TRUE

TRUE

ESC

FALSE

OK

mixer mode

auto		ESC OK
auto		
manual		
clean		
fault		

sterilization time

▲

▲

5

0

▼

▼

MINUTE

SEC

ESC

OK

3.4 Übersetzung

Der Datamanager nutzt den CODESYS-Übersetzungsmechanismus für Gruppenbezeichnungen, Wertenamen und Einheiten. Die Übersetzung ist clientspezifisch konfigurierbar. Zusätzlich ist ein Mechanismus für die Teiltextauflösung integriert. Um einen Teilstring zu übersetzen, wird er der jeweiligen Gruppe per `grp.AddKey()` hinzugefügt.

3.4.1 Beispiel Teiltextauflösung

Sollen die Bezeichnungen „STERILIZATION“ und „CYCLE“ übersetzt werden, genügt ein einziger Eintrag von „STERILIZATION“ und „CYCLE“ in der Übersetzungsdatei, um alle Strings mit dem Namen zu übersetzen.

ID	Default	de
STERILIZATION	sterilization	Sterilisierung
CYCLE	cycle	Zyklus

```

Steril.InitGroup(
    // Group - Sterilization
    sName      := 'STERILIZATION CYCLE',           // Group Name
    sTranslationFile := 'txtTestTranslations',       // [] Translation file name
    parent     := Linel);                          // Parent Group
Steril.AddKey('STERILIZATION');
Steril.AddKey('CYCLE');

SterilTemp.Init(
    //Value - Sterilization Temperature
    group      := Steril,
    sName      := 'STERILIZATION TEMP',
    fMin       := 131,
    fDefault   := 185,
    fMax       := 275,
    sUnit      := 'FAH',
    nDecimals  := 2);

SterilTime.Init(
    //Value - Sterilization Time
    group      := Steril,
    sName      := 'STERILIZATION TIME',
    nDefault   := DAM.UTL.TimeFormat.Merge(nMin := 5),
    eFormat    := DAM.UTL.TIME_FORMAT.MINUTE_SEC);

```



HINWEIS: ASKS GmbH verwendet ausschließlich Englisch als Standard-ID. Großbuchstaben-IDs machen nicht übersetzte Einträge sofort erkennbar.

3.5 Einheitenkonvertierung

Der REAL-Wert des Datamanagers verfügt über integrierte Funktionen zur Einheitenkonvertierung. Verschiedene Einheiten sind vordefiniert und in der AS_ System-Bibliothek hinterlegt. Auf Kundenwunsch können weitere Einheiten ergänzt werden.

3.5.1 Beispiel Einheitenkonvertierung

In einer Applikation wird ein Temperaturwert in Fahrenheit verwendet, im europäischen Markt ist Celsius gebräuchlicher. Die Konvertierung wird als Übersetzungsstring im Format BASE<>TARGET definiert, wobei BASE die Ausgangs- und TARGET die Zieleinheit bezeichnet.

ID	Default	de
FAH	FAH	FAH<>CEL

```

SterilTemp.Init(
    group           := Steril,
    sName           := 'STERILIZATION TEMP',
    fMin            := 131,
    fDefault        := 185,
    fMax            := 275,
    sUnit           := 'FAH',
    nDecimals       := 2);
//Value - Sterilization Temperature
    
```

Sobald auf die entsprechende Sprache umgeschaltet wird, konvertiert der Datamanager den Wert automatisch.

recipe management/production line 1/sterilization cycle			
Group	Value	Actual	Unit
	sterilization temp	185.0	FAH
	sterilization time	5:00	mm:ss

de

en

sterilization temp

131.0

275.0

185.0

7	8	9	Back
4	5	6	Clear
1	2	3	ESC
0	+/-	.	OK

Rezepte Verwaltung/Produktions Linie 1/Sterilisierung Zyklus			
Group	Value	Actual	Unit
	Sterilisierung Temp	85.0	CEL
	Sterilisierung Zeit	5:00	mm:ss

de

en

Sterilisierung Temp

55.0

135.0

85.0

7	8	9	Back
4	5	6	Clear
1	2	3	ESC
0	+/-	.	OK

4 Lizenz

Informationen zur Lizenzierung dieser Bibliothek erhalten Sie bei ASKS GmbH. Bitte kontaktieren Sie uns unter support@as-ks.de oder über unsere Website www.as-ks.de.