



Dynamische Speicherzuweisung

FAQ No.0006

Part	Version	Revision	Date	Status
de	6.2.3.0	002	2020-03-12	Released

Inhalt

Einführung	2
__NEW, __DELETE (CoDeSys RTE V3.x)	2
SysMemAllocData, SysMemFreeData (CoDeSys RTE V3.x)	4
SysMemAllocData	4
SysMemFreeData	4
Beispiel.....	5
Disclaimer	6

Einführung

Dieses Dokument gibt eine kleine Einführung in die dynamische Speicherzuweisung auf einem C6 Econ/Perform.

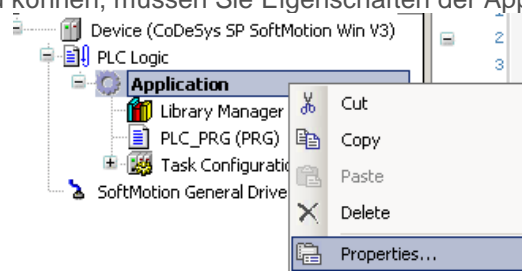
Beachten Sie: Speicherzuweisung kann zu hohem Jitter und langen Zykluszeiten führen (am Besten in Initialisierungsschritten verwenden).

__NEW, __DELETE (CoDeSys RTE V3.x)

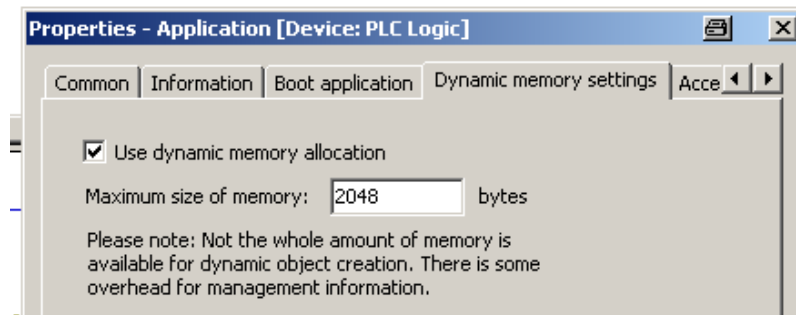
Die CoDeSys RTE V3 unterstützt einige Funktionen zur dynamischen Speicherverwaltung, die nicht Teil der Norm IEC 61131-3 sind.

Diese Methode funktioniert nicht mit der C6 Compact.

Um diese Funktion nutzen zu können, müssen Sie Eigenschaften der Applikation ändern.



Wählen Sie dann, wie viel Speicher Sie verwenden wollen:



Beachte: Wenn Sie das Gerät aktualisieren, gehen diese Einstellungen verloren!

Nutzen Sie den __NEW Operator, um den benötigten Speicherplatz zuzuweisen.

Wenn Sie den Speicher nicht mehr brauchen, vergessen Sie nicht, den Speicher mit __DELITE freizugeben.

Hier ist ein Beispiel, um ein Array mit insgesamt 25 Bytes zu erstellen.

FAQ COMBIVIS studio 6



```
PLC_PRG
1 PROGRAM PLC_PRG
2 VAR
3     bInit: BOOL := TRUE;
4     bDelete: BOOL;
5     pArrayBytes : POINTER TO BYTE; //Pointer to Byte (will be pointing to your array after allocation)
6     test: BYTE; //Test-variable to check for function
7 END_VAR

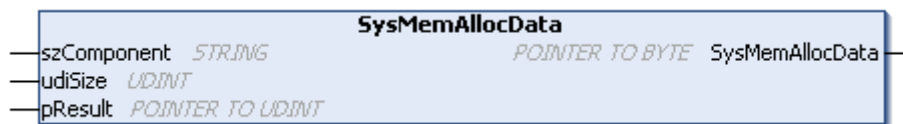
1 IF (bInit) THEN
2     pArrayBytes := __NEW(BYTE, 25); //This creates a new array of type Byte and 25 Elements
3     bInit := FALSE;
4 END_IF
5
6 IF (pArrayBytes <> 0) THEN //Pointer will be 0 if creation fails
7     pArrayBytes[24] := 125; //Some sample code
8     test := pArrayBytes[24];
9 END_IF
10
11 IF (bDelete) THEN
12     __DELETE(pArrayBytes); //When variable is not needed anymore dont forget to free the memory
13 END_IF
```

SysMemAllocData, SysMemFreeData (CoDeSys RTE V3.x)

Diese Version verwendet die SysMem-Systembibliothek von CoDeSys. Es gibt viele Funktionen, um verschiedene Arten von Daten für verschiedene Arten von Speicher zuzuordnen und freizugeben. Die Methode funktioniert bei der C6 Compact nicht.

In dieser Anleitung werden nur die SysMemAllocData und SysMemFreeData erläutert.

SysMemAllocData



- | | |
|----------------------|--|
| szComponent (STRING) | Geben Sie den Speicherbereich, den Sie zuweisen möchten, einen Namen. Dieser Name wird später verwendet, um den Speicher freizugeben |
| udiSize (UDINT) | Die Größe in BYTES, die Sie zuweisen möchten |
| pResult (PTR) | Error Code (0 = kein Error) |
| SysMemAllocData | Liefert einen Zeiger, wenn der Speicher korrekt zugewiesen wurde. |

SysMemFreeData



- | | |
|----------------------|---|
| szComponent (STRING) | Verwenden Sie den gewählten Namen mit SysMemAllocData |
| pMemory (PTR) | Zeiger auf den Zugeordneten Speicher |
| SysMemFreeData | Error Code (0 = kein error) |

Beispiel

Erstellung eines INT-Array:

```
1  PROGRAM PLC_PRG
2  VAR
3      res: RTS_IEC_RESULT;           //IEC result
4      pdata: POINTER TO INT;        //Pointer to int (creation of int array)
5
6      i:INT;
7      bInit:BOOL:=TRUE;
8      bErase:BOOL:=FALSE;
9  END VAR

```

```
1  IF bInit THEN
2      pdata:=SysMemAllocData('a', SIZEOF(INT)*25, ADR(res)); //allocate 25 INT and assign it to pdata
3      IF res = 0 THEN //0 if allocation complete without error
4          bInit:=FALSE;
5      END_IF
6  END_IF
7
8  FOR i:=0 TO 24 BY 1 DO //Some sample code
9      pdata[i]:=i+900; //Use it like a normal array
10 END_FOR
11
12 IF bErase THEN
13     SysMemFreeData('a', pdata); //free memory
14 END_IF

```



Disclaimer

KEB Automation KG reserves the right to change/adapt specifications and technical data without prior notification. The safety and warning reference specified in this manual is not exhaustive. Although the manual and the information contained in it is made with care, KEB does not accept responsibility for misprint or other errors or resulting damages. The marks and product names are trademarks or registered trademarks of the respective title owners.

The information contained in the technical documentation, as well as any user-specific advice in verbal or in written form are made to the best of our knowledge and information about the application. However, they are considered for information only without responsibility. This also applies to any violation of industrial property rights of a third-party.

Inspection of our units in view of their suitability for the intended use must be done generally by the user. Inspections are particular necessary, if changes are executed, which serve for the further development or adaption of our products to the applications (hardware, software or download lists). Inspections must be repeated completely, even if only parts of hardware, software or download lists are modified.

Application and use of our units in the target products is outside of our control and therefore lies exclusively in the area of responsibility of the user.

KEB Automation KG
Südstraße 38 • D-32683 Barntrup
fon: +49 5263 401-0 • fax: +49 5263 401-116
net: www.keb.de • mail: info@keb.de