Dynamic memory allocation

FAQ No.0006

Part	Version	Revision	Date	Status
en	6.2.3.0	001	2019-01-01	Released

Content

KEB Automation KG
Südstraße 38
32683 Barntrup, Germany
E-mail: info@keb.de
Tel. +49 5263 401-0
Fax +49 5263 401-116
www.keb.de

District Court Lemgo HRA 5649
DUNS-No. 314108728
VAT-No. DE309087075
Bank Details: Sparkasse Paderborn-Detmold
IBAN DE 19 4765 0130 0000 0060 07
BIC WELADE3L

General Partner:
Vittorio Tavella
KEB Verwaltungs-GmbH, Barntrup
District Court: Lemgo HRB 8965
Directors: Curt Bauer CMO,
Ralf Lutter COO, Vittorio Tavella CFO,
Wolfgang Wiele CTO

Introduction

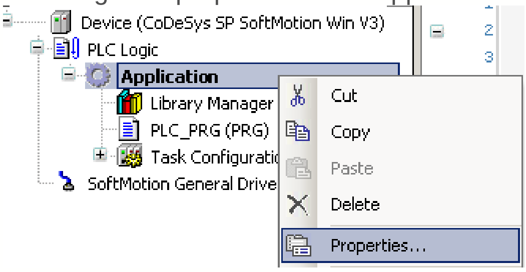
This document gives a little introduction to dynamic memory allocation on C6 Econ/Perform.

Note: Memory allocation can cause high jitter and long cycle times (Best to use in initialization steps).

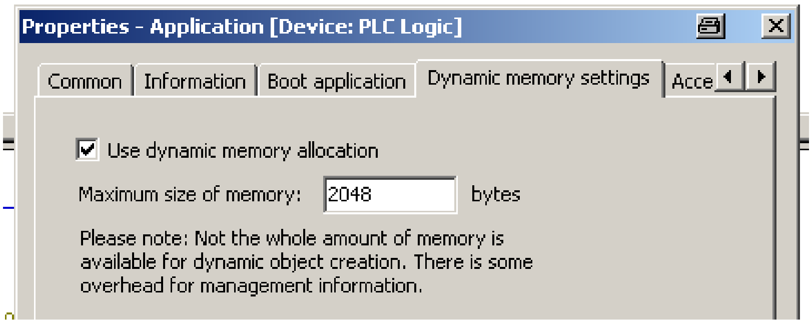
__NEW, __DELETE (CoDeSys RTE V3.x)

The CoDeSys RTE V3 supports some functions for dynamic memory management which are NOT part of the IEC 61131-3 standard.
This method is not working for C6 Compact.

To use this features you have to change the properties of the application.



Then choose how much memory you want to use:

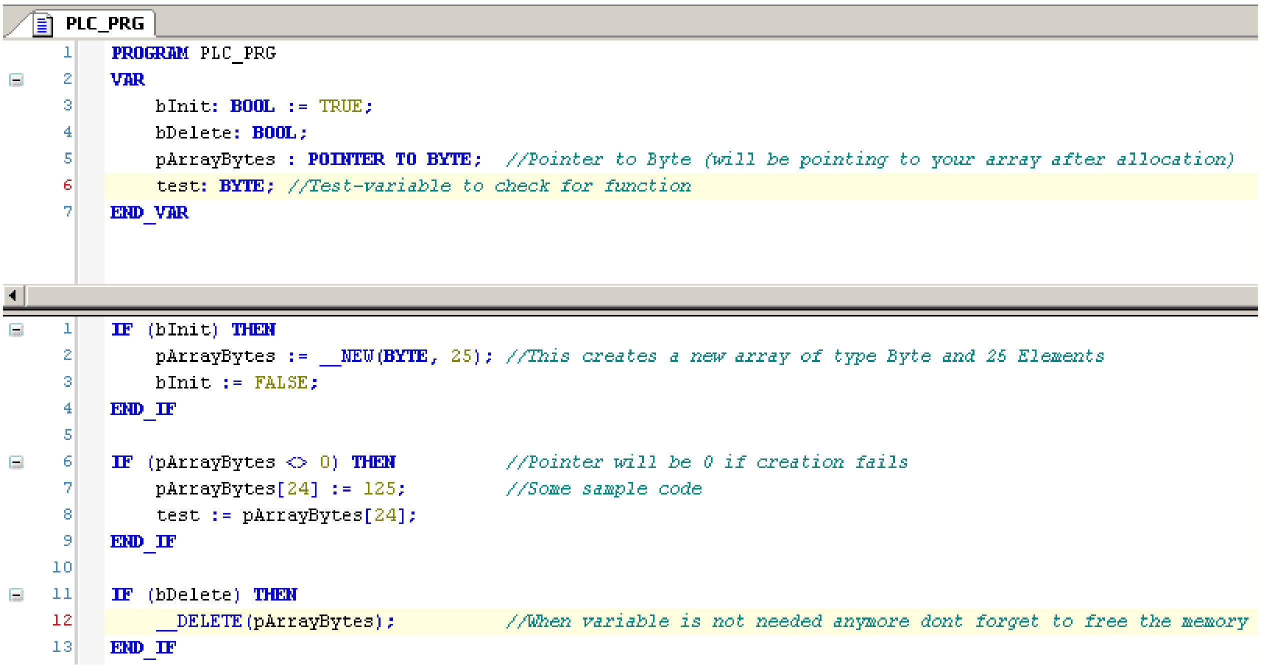


☑ Use dynamic memory allocation

Maximum size of memory: 2048 bytes

Please note: Not the whole amount of memory is available for dynamic object creation. There is some overhead for management information.

Note: When you update the Device this settings will be lost!
Use the __NEW Operator to allocate the memory you need.
When you don’t need the memory anymore do not forget to free the memory with __DELETE!
Here is a sample to create an Array of 25 bytes total:



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

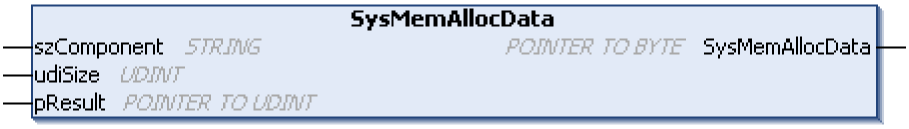
```
IF (bInit) THEN
    pArrayBytes := __NEW(BYTE, 25); //This creates a new array of type Byte and 25 Elements
    bInit := FALSE;
END_IF

IF (pArrayBytes <> 0) THEN          //Pointer will be 0 if creation fails
    pArrayBytes[24] := 125;         //Some sample code
    test := pArrayBytes[24];
END_IF

IF (bDelete) THEN
    __DELETE(pArrayBytes);          //When variable is not needed anymore dont forget to free the memory
END_IF
```

SysMemAllocData, SysMemFreeData (CoDeSys RTE V3.x)

This version is using the SysMem-system library from CoDeSys. There are many functions to allocate and free different types of data for different types of memory.
This method is not working for C6 Compact.

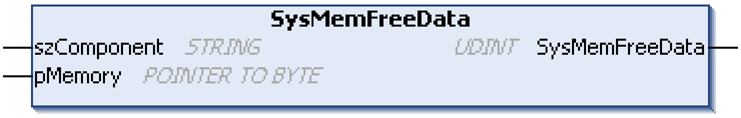
In this How-to only the SysMemAllocData and SysMemFreeData will be explained.

SysMemAllocData



szComponent (STRING)	Set a name to the memory area you want to allocate. This name is used later to free the memory
udiSize (UDINT)	The size in BYTES(!) you want to allocate
pResult (PTR)	Error Code (0 means no error)
SysMemAllocData	Returns a Pointer when memory is allocated properly

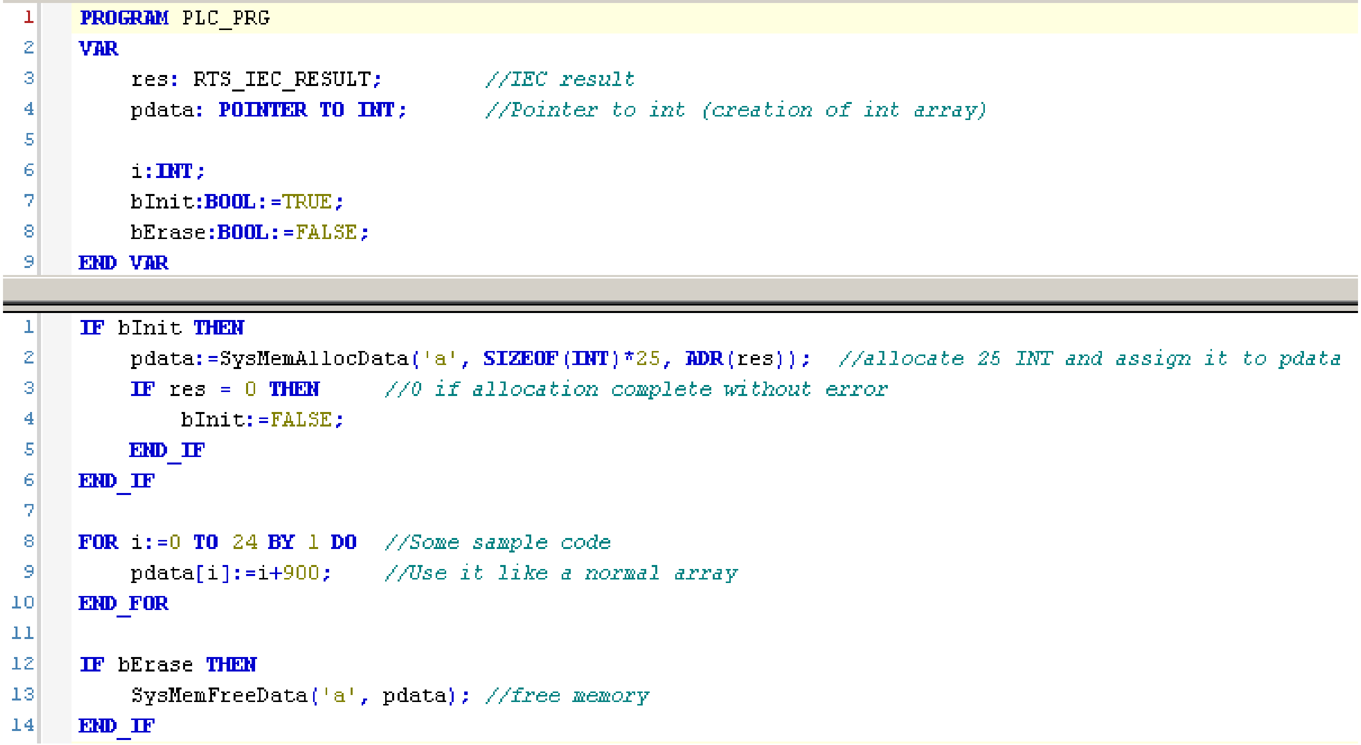
SysMemFreeData



szComponent (STRING)	Use the name chosen with SysMemAllocData
pMemory (PTR)	Pointer to the allocated memory
SysMemFreeData	Error Code (0 means no error)

Sample

Creation of INT-Array:



```
PROGRAM PLC_PRG
VAR
    res: RTS_IEC_RESULT;        //IEC result
    pdata: POINTER TO INT;      //Pointer to int (creation of int array)

    i:INT;
    bInit:BOOL:=TRUE;
    bErase:BOOL:=FALSE;
END_VAR
```

```
IF bInit THEN
    pdata:=SysMemAllocData('a', SIZEOF(INT)*25, ADR(res));  //allocate 25 INT and assign it to pdata
    IF res = 0 THEN     //0 if allocation complete without error
        bInit:=FALSE;
    END_IF
END_IF

FOR i:=0 TO 24 BY 1 DO  //Some sample code
    pdata[i]:=i+900;    //Use it like a normal array
END_FOR

IF bErase THEN
    SysMemFreeData('a', pdata); //free memory
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