

SIEMENS



FAQ • 10/2015

Storage Concept and Data Backup Mechanisms for SIMATIC S7-300

SIMATIC S7-300

<https://support.industry.siemens.com/cs/ww/en/view/109478260>

This entry originates from the Siemens Industry Online Support. The conditions of use specified there apply (www.siemens.com/nutzungsbedingungen).

Security Notes

Siemens offers products and solutions with industrial security functions which support the secure operation of plants, solutions, machines, devices and/or networks. They are important components in a comprehensive industrial security concept. The Siemens products and solutions continue to be developed under this aspect. Siemens recommends that you keep yourself regularly informed about product updates.

For the safe operation of Siemens products and solutions it is necessary to take appropriate security measures (cell protection concept, for example) and to integrate each component in an overall industrial security concept which is state of the art. This should also cover the third-party products used. Additional information about industrial security is available at:

<http://www.siemens.com/industrialsecurity>.

In order to keep yourself informed about product updates, we recommend subscribing to our product-specific newsletter. Additional information about this is available at <http://support.industry.siemens.com>.

Contents

1	Introduction	3
2	Saving the Actual Values in the Offline Project	4
3	Copying Data from RAM to ROM	6
4	Saving and Read-back of Actual Values with System Functions.....	8
5	Setting Data to Retentive	10

1 Introduction

During commissioning, certain values in the data blocks are changed for the plant. These can be control parameters, limit values, etc. These changes are often made directly in the CPU, in other words online. In order to retain these changes, the data has to be saved beforehand.

This FAQ response illustrates the storage concept of a SIMATIC S7-300 CPU and describes the options for saving data and actual values so that they are retained after POWER OFF and restart.

Furthermore there is a description of how to save data before an overall reset and reloading of the program.

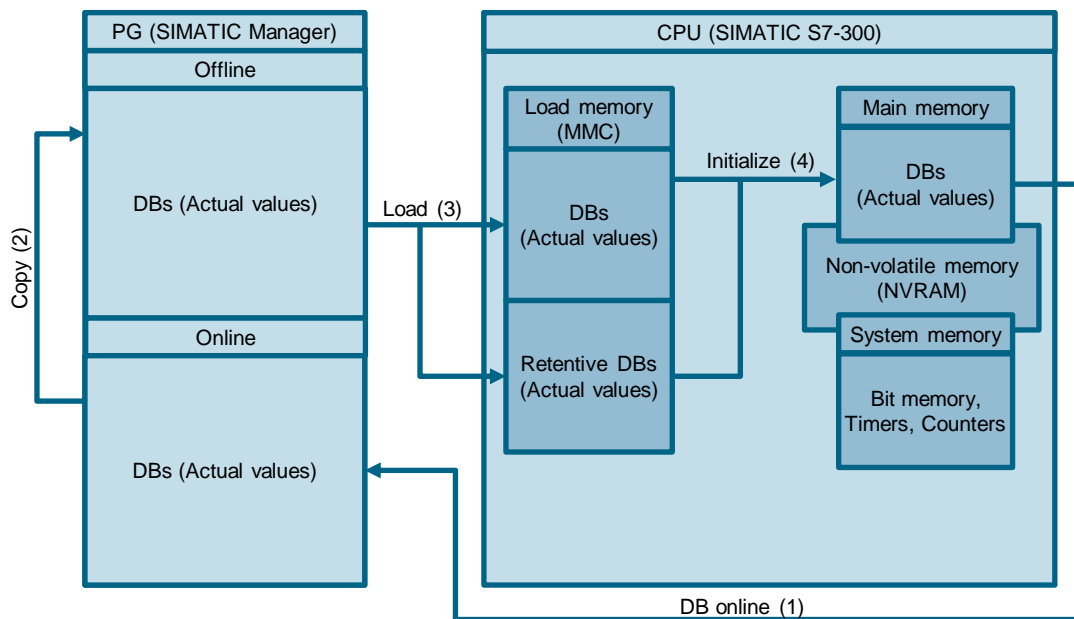
2 Saving the Actual Values in the Offline Project

You can save actual values of data blocks by copying the online blocks to your offline project and loading the blocks once again into the SIMATIC S7-300.

This reloading means that this data is once again available in the load memory and thus also after an overall reset.

Schematic diagram of saving actual values

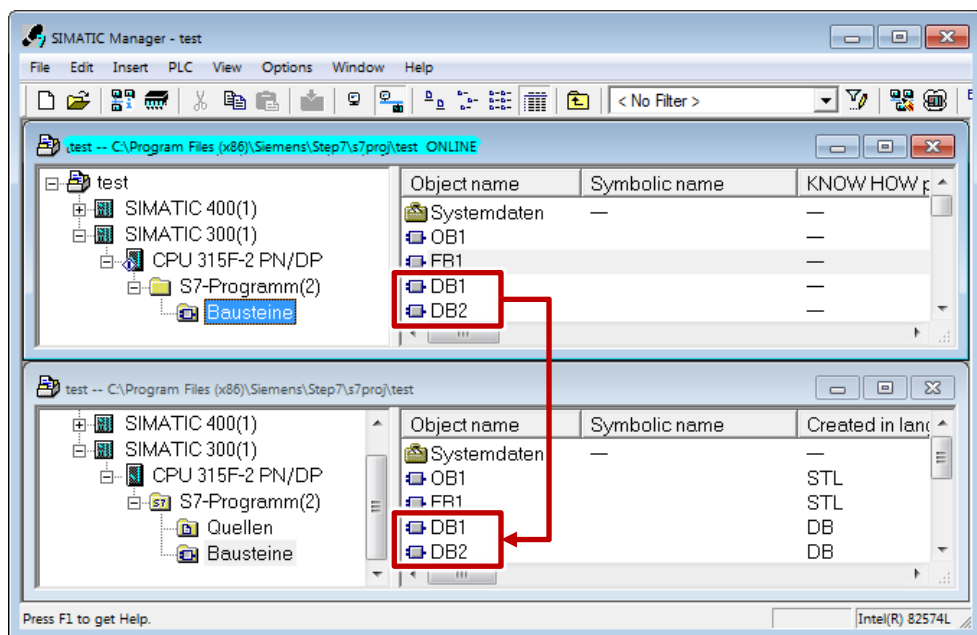
Figure 2-1



Procedure for backing up the actual values

1. In the SIMATIC Manager you open the dialog "Open Project" by selecting the menu command "File > Open..." and then selecting the project whose program is loaded in the CPU.
2. Then in the SIMATIC Manager you open the Online view with the menu command "View > Online". For a better overview we recommend tiling the windows one on top of the other (Online and Offline views) by selecting the menu command "Window > Arrange > Tile Horizontally".
3. Now you can drag and drop or copy the data blocks from the Online view to the Offline view. This then transfers the actual values to the Offline project.

Figure 2-2



Note Make sure that the online blocks and offline blocks have the same time stamp; otherwise the offline declaration view is lost!

Note If a DB block from the online project folder is opened and there is a difference in time stamp (interface, start values), then dummy variables are also displayed (STAT0, for example) for the variables.

Note If the blocks loaded by the controller into the project are loaded back into the controller, the last actual values become the new start values.

3 Copying Data from RAM to ROM

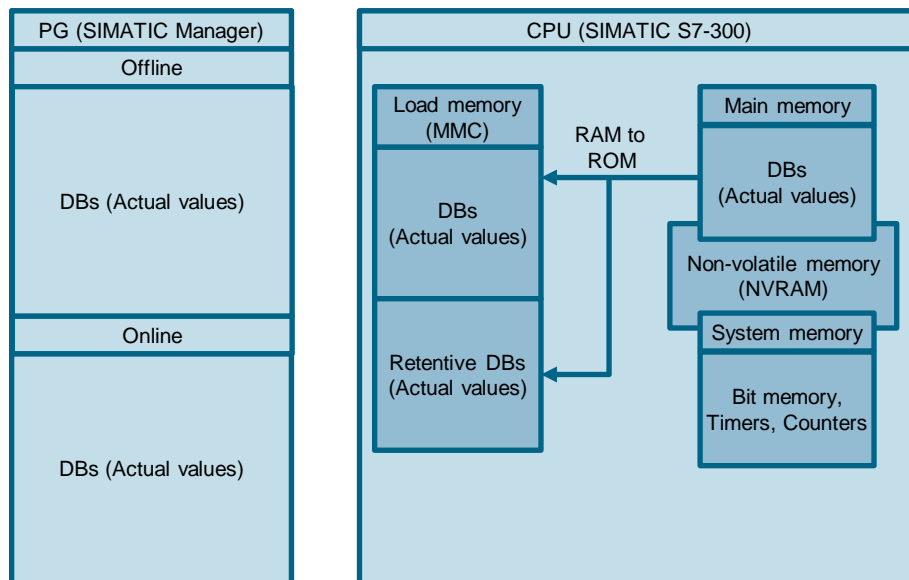
With the "Copy RAM to ROM..." function the actual values of the runtime-relevant retentive and non-retentive data blocks in the load memory are overwritten with the actual values from the main memory.

These actual values remain available after an overall reset, because after an overall reset the actual values are transferred from the load memory to the main memory of the SIMATIC S7-300.

Note Note that the actual values in the load memory are overwritten when the program is loaded.

Schematic diagram of copying data from RAM to ROM

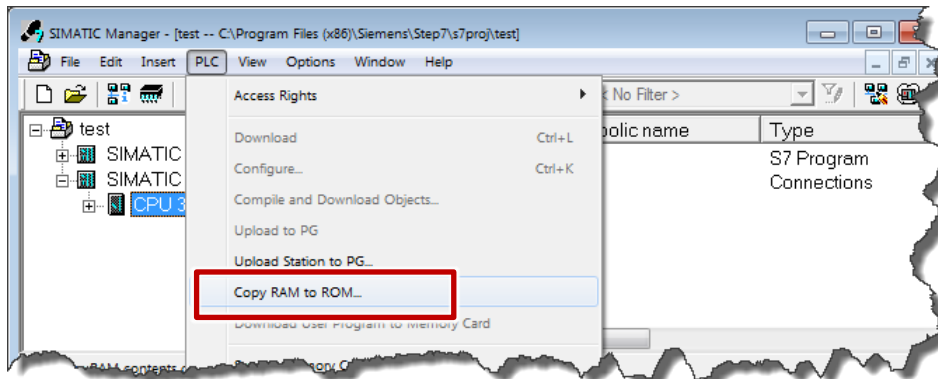
Figure 3-1



Procedure for copying data from RAM to ROM

1. Open the project in the SIMATIC Manager and mark the controller.
2. Select the menu command "PLC > Copy RAM to ROM...".

Figure 3-2



Note The function is only permitted when the CPU is in STOP mode.

Note The "Copy RAM to ROM..." function only works for data blocks that are **not** designated as "unlinked" (DB only in the load memory).

Note If the function could not be terminated with power off, the load memory is then empty. From the check sum the CPU recognizes that the data has not been completely written. The program is recognized as invalid and has to be reloaded. In this case the data blocks are overwritten with the new start values in the load memory.
--> It is therefore useful to first copy the online blocks into the offline project.

Note No program blocks are copied to the load memory!

4 Saving and Read-back of Actual Values with System Functions

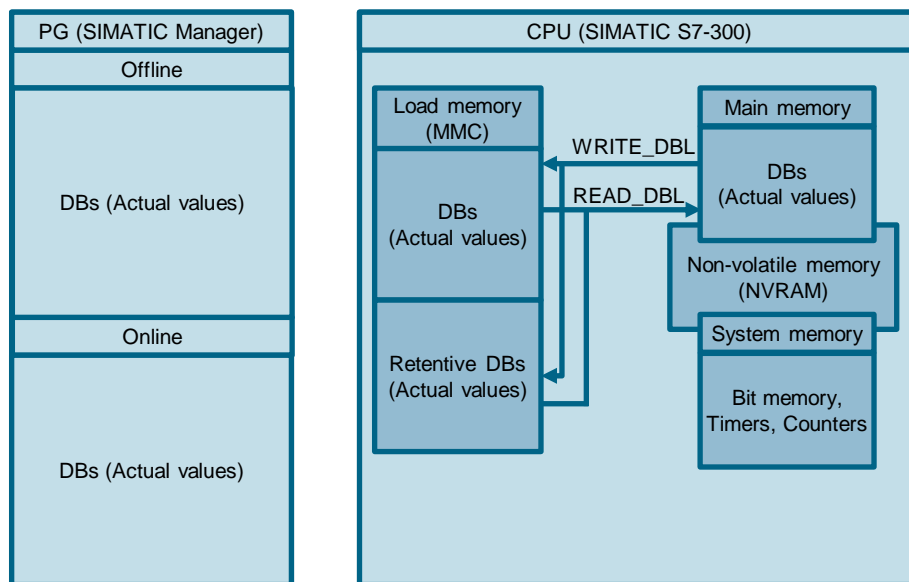
Using the system functions SFC 84 "WRIT_DBL" and SFC 83 "READ_DBL" you can write the actual values from the main memory to the load memory (Micro Memory Card) and read them back again. In this way you can save the data of retentive and non-retentive data blocks before an overall reset.

Note

Note that the actual values in the load memory are overwritten when the program is loaded.

Schematic diagram for saving and read-back of actual values with system functions

Figure 4-1



Call of system functions for saving and read-back of actual values

With system function SFC 84 "WRIT_DBL" (write data block in load memory) you save the data areas from the main memory in the load memory (Micro Memory Card).

Figure 4-2

```
CALL "WRIT_DBL"           SFC84
REQ    :=M0.0
SRCBLK :=DB2
RET_VAL:=#retval         #retval
BUSY   :=#busy          #busy
DSTBLK :=DB2
```

Note SFC 84 is **not** suitable for writing variables frequently (or cyclically) to the load memory, because the micro memory card technology only permits a specific number of write accesses (100,000 write/delete operations per cell).

With system function SFC 83 "READ_DBL" (read data block in load memory) you copy the data area as required back from the load memory into the main memory.

Figure 4-3

```
CALL "READ_DBL"          SFC83
REQ    :=M0.1
SRCBLK :=DB2
RET_VAL:=#retvalread    #retvalread
BUSY   :=#busyread      #busyread
DSTBLK :=DB2
```

Note After "STOP/RUN", "Power off / power on" and "Overall reset" the data is transferred automatically to the main memory. It is therefore not necessary to read the data with the system function SFC 83 "READ_DBL".

5 Setting Data to Retentive

In the configuration you define which parts are to be retentive and which are to be initialized with "0" with a restart (warm start).

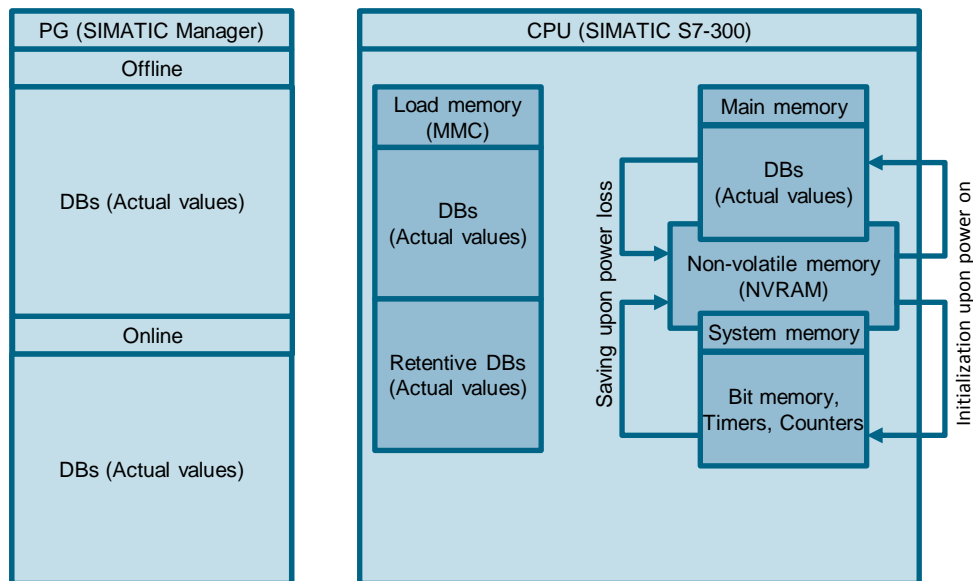
If there is a power failure in the SIMATIC S7-300 or in the transition of the SIMATIC S7-300 from RUN to STOP, the retentive data is saved from the main memory (RAM) to a separate memory area of the CPU. The main memory (RAM) is "empty" after a power failure. Upon restart the program blocks and the non-retentive data blocks are loaded from the load memory into the main memory (RAM). The retentive data blocks are loaded from the separate memory area of the CPU.

Note

Note that with an overall reset the retentive area is also deleted. Then the retentive data blocks are also loaded from the load memory into the main memory (RAM). Data which has been changed by the process, for example, or the HMI, is also initialized through the values of the load memory.

Schematic diagram of setting data to retentive

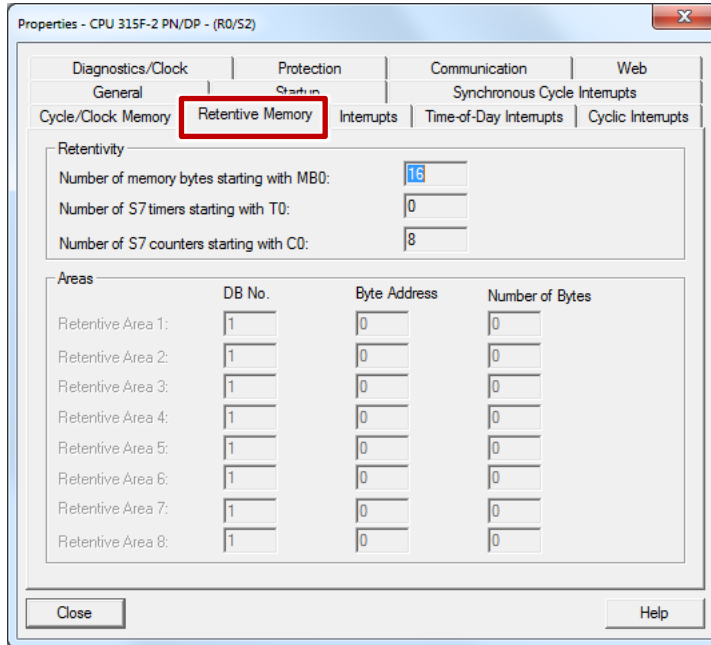
Figure 5-1



Procedure for setting system memory elements to retentive

The system memory (RAM) includes memory elements like markers, timers and counters. In the configuration under "Properties of the CPU > Retentive Memory tab" you define for these elements which parts are to be retentive and which are to be initialized with "0" with a restart (warm start).

Figure 5-2



Procedure for setting data in data blocks to retentive

Data blocks are created by default as retentive. With the "Non Retain" property you can also create non-retentive data blocks.

Figure 5-3

