SIEMENS

SIMATIC
Industrial software
Readme SIMATIC S7-PLCSIM Advanced V4.0
Readme

General information

Content

This Readme file contains information about SIMATIC S7-PLCSIM Advanced V4.0.

This information should be considered more up-to-date than the product documentation and, where necessary, replaces corresponding information in the product documentation and installation instructions.

Amendment to section 2.7 Password to protect confidential configuration data

Editing an instance created with PLCSIM Advanced V4.0 using PLCSIM Advanced API V3.0

You have assigned a password to protect confidential configuration data for the real CPU.

Note

Editing an instance created with PLCSIM Advanced V4.0 using PLCSIM Advanced API V3.0 with the functions ArchiveStorage() and RetrieveStorage()

When you edit an instance created with PLCSIM Advanced V4.0 with the API functions ArchiveStorage() and RetrieveStorage() of V3.0, API V3.0 does not save the entire card content of the virtual memory card. The retain.pms and sim_hwdb.ini files are missing.

This means that the password to protect confidential configuration data is not saved and will be lost.

Remedies

You have a variety of remedy options:

- Set the password to protect the confidential configuration data before each simulation again using STEP 7, the SIMATIC Automation Tool or Openness.
- Establish a remote connection of the instance via the Runtime Manager even when the instance communicates locally. In this case, the password is saved to the virtual memory card.
- Do not assign a password to protect confidential configuration data.

Simultaneous download of an F-CPU and a standard CPU to the same instance

You have created an instance using the Control Panel. You have not assigned a password to protect confidential configuration data in STEP 7.

In addition to a standard CPU, you also want to load an F-CPU via Softbus or TCP/IP to an instance.

When you try to load an F-CPU to the same instance to which you have already loaded a standard CPU, an error is displayed with the request to enter a password.

Remedy

1. Create a STEP 7 project with a standard CPU and an F-CPU of the same performance class, e.g., CPU 1518-4 PN/DP and CPU 1518F-4 PN/DP.
2. Load the standard CPU to the instance.
3. Load the F-CPU to the same instance.

   Result: You are prompted to enter a password via a window.
4. Cancel the download by clicking on the "Cancel" button.
5. Load the F-CPU again to the instance.

   Result: The download is performed.

Amendment to section 6.3.1 Synchronize simulation partner cycle-controlled

Correction for Table 6-1 Cycle-controlled operating modes (SingleStep)

In contrast to the specifications in the S7-PLCSIM Advanced Function Manual, note the **footnotes** [2] **and** [3]:

Table 1 Cycle-controlled operating modes (SingleStep)

Operating mode	Synchronization point		Minimum cycle time[1]	Send clock "Bus"[2]
	Cycle control point	Before reading in the process image partition		
	"C"	"P"	"T"	
`SingleStep_C`	✓			
`SingleStep_P`		✓		
`SingleStep_CP`	✓	✓		
`SingleStep_CT`	✓		✓	
`SingleStep_CPT`	✓	✓	✓	
`SingleStep_Bus`[3]				✓

[1] In addition, the minimum scan cycle time of the OB 1 is overwritten in this operating mode. When you define a minimum scan cycle time of 200 ms via the API, the minimum interval between two cycle control points is 200 virtual milliseconds. The default setting is 100 ms.

[2] Send clock of the IO system (PROFIBUS or PROFINET) that is to be used for the cycle-controlled synchronization of the virtual controller. You set the send clock of the respective IO system in the STEP 7 properties, for example, in the properties of the PROFINET interface of the CPU (Advanced options > Realtime settings > IO communication > Send clock).

[3] Conditions for the `SingleStep_Bus` operating mode:
- You must have configured outputs at the controller.
- You must call the SYNC_PO instruction in the isochronous mode interrupt OB to synchronize the process image of the outputs.

Amendment to section 7.6.3 Operating state

RUN()

In contrast to the information included in the S7-PLCSIM Advanced Manual, the RUN() function can return an additional return value:

Table 2 RUN() - Native C++ and .NET (C#)

Return values	Runtime error code	Conditions
	`SREC_IS_EMPTY`	One of the following two conditions applies here: **Condition:** After creating and starting an instance, you have tried to switch the virtual controller to the RUN operating state without downloading a STEP 7 project. **Condition:** After creating and enabling an instance with download of a STEP 7 project with enabled password for the protection of confidential configuration data, you have moved the virtual memory card from one system to another. You have tried to start the instance on the new system. The following error code is returned after the start: `"SREC_WARNING_PASSWORD_PROTECTION_ERROR"` Afterwards, you have tried to switch the virtual controller to the RUN operating state.

RUN operating state after enabling an instance

When you expect the RUN operating state after enabling an instance, check whether the instance is already in RUN by subscribing to the event `OnOperatingStateChanged()`. As an alternative, call `Run()` immediately after enabling the API function to switch the instance to the RUN operating state.

Executing PowerOff()

After its start, the API function `PowerOff()` returns the error code `"-39 InvalidOperatingState"`.

Remedy

Before you execute the API function `PowerOff()`, you must cancel the subscriptions to all events; otherwise, you will receive unexpected exceptions.

Amendment to section 7.6.7 Cycle control

RunToNextSyncPoint()

In contrast to the specifications in the S7-PLCSIM Advanced Function Manual, note the following updated tables:

Table 3 RunToNextSyncPoint() - Native C++

Syntax	`ERuntimeErrorCode RunToNextSyncPoint();`
	`ERuntimeErrorCode RunToNextSyncPoint(UINT32 in_IoSystemId, UINT64 in_BusTimeStamp_ns);`	

Parameters	The call with parameters is only valid for the `SingleStep_Bus` operating mode. • `UINT32 in_IoSystemId:` Number of the IO system (PROFIBUS 1..32, PROFINET 100..115) that is to be used for the cycle-controlled synchronization of the virtual controller. • `UINT64 in_BusTimeStamp_ns:` Time stamp in nanoseconds of the respective IO system at the start of each cycle.	
Return values	Runtime error code	Condition
	`SREC_OK`	The function is successful.
	`SREC_INTERFACE_REMOVED`	The instance is not registered in Runtime Manager.
	`SREC_TIMEOUT`	The function does not return on time.
	`SREC_INSTANCE_NOT_RUNNING`	The process of the virtual controller is not running.
	`SREC_WRONG_ARGUMENT`	For the `SingleStep_Bus` operating mode: The ID of the IO system is invalid. For all other operating modes: The function call with parameters is invalid.

Table 4 RunToNextSyncPoint() - .NET (C#)

Syntax	`void RunToNextSyncPoint();`
	`void RunToNextSyncPoint(UINT32 in_IoSystemId, UINT64 in_BusTimeStamp_ns);`	

Parameters	The call with parameters is only valid for the `SingleStep_Bus` operating mode. • `UINT32 in_IoSystemId:` Number of the IO system (PROFIBUS 1..32, PROFINET 100..115) that is to be used for the cycle-controlled synchronization of the virtual controller. • `UINT64 in_BusTimeStamp_ns:` Time stamp in nanoseconds of the respective IO system at the start of each cycle.	
Return values	None	
Exceptions	`Siemens.Simatic.Simulation.Runtime.SimulationRuntimeException`	
	Runtime error code	Condition
	`ERuntimeErrorCode.InterfaceRemoved`	The instance is not registered in Runtime Manager.
	`ERuntimeErrorCode.Timeout`	The function does not return on time.
	`ERuntimeErrorCode.NotRunning`	The process of the virtual controller is not running.
	`ERuntimeErrorCode.WrongArgument`	For the `SingleStep_Bus` operating mode: The ID of the IO system is invalid. For all other operating modes: The function call with parameters is invalid.

Amendment to section 7.8.2.6 EventCallback_II_SREC_ST_UINT32_INT64_UINT32

Description

In contrast to the specifications in the S7-PLCSIM Advanced Function Manual, note the following updated table:

Table 5 EventCallback_II_SREC_ST_UINT32_INT64_INT64_UINT32 - Native C++

Syntax	`typedef void (*EventCallback_II_SREC_ST_UINT32_INT64_INT64_UINT32)(`
	` IInstance* in_Sender,`
	` ERuntimeErrorCode in_ErrorCode,`
	` SYSTEMTIME in_SystemTime,`
	` UINT32 in_Id,`
	` INT64 in_TimeSinceSameSyncPoint_ns,`
	` INT64 in_TimeSinceAnySyncPoint_ns,`
	` UINT32 in_SyncPointCount`
	`);`

Parameters	• `IInstance* in_Sender:` An interface of the instance that receives this event. • `ERuntimeErrorCode in_ErrorCode:` A possible error code. • `SYSTEMTIME in_SystemTime:` The virtual system time of the virtual controller at the time when this event was triggered. • `UINT32 in_Id:` For the `SingleStep_Bus` operating mode: The ID of the IO system that is used for cycle-controlled synchronization of the virtual controller. For all other operating modes: The ID of the process image partition (PIP) that triggers this event. 0 for the cycle control point (End of cycle). • `INT64 in_TimeSinceSameSyncPoint_ns:` The virtual time (in nanoseconds) since the last synchronization point of the same process image partition ID was reached. For the time-controlled operating modes: The runtime since the last call of the `StartProcessing()` function. • `INT64 in_TimeSinceAnySyncPoint_ns:` The virtual time (in nanoseconds) since the last synchronization point of any process image partition ID was reached. For the time-controlled operating modes: The runtime since the last call of the `StartProcessing()` function. • `UINT32 in_SyncPointCount:` The number of synchronization points since the last event. If the events are triggered faster than they are received, multiple events are combined into one event. In this case, this value contains the number of cycles since the last event was received.
Return values	None

Amendment to section 7.8.3.7 Delegate_II_EREC_DT_UINT32_INT&$_INT64_UINT32

Description

In contrast to the specifications in the S7-PLCSIM Advanced Function Manual, note the following updated table:

Table 6 Delegate_II_EREC_DT_UINT32_INT64_INT64_UINT32 - .NET (C#)

Syntax	`delegate void Delegate_II_EREC_DT_UINT32_INT64_INT64_UINT32 (`
	` IInstance in_Sender,`
	` ERuntimeErrorCode in_ErrorCode,`
	` DateTime in_DateTime,`
	` UInt32 in_Id,`
	` Int64 in_TimeSinceSameSyncPoint_ns,`
	` Int64 in_TimeSinceAnySyncPoint_ns,`
	` UInt32 in_SyncPointCount`
	`);`

Parameters	• `IInstance in_Sender:` An interface of the instance that receives this event. • `ERuntimeErrorCode in_ErrorCode:` A possible error code. • `DateTime in_DateTime:` The virtual system time of the virtual controller at the time when this event was triggered. • `UInt32 in_Id:` For the `SingleStep_Bus` operating mode: The ID of the IO system that is used for cycle-controlled synchronization of the virtual controller. For all other operating modes: The ID of the process image partition (PIP) that triggers this event. 0 for the cycle control point (End of cycle). • `Int64 in_TimeSinceSameSyncPoint_ns:` The virtual time (in nanoseconds) since the last synchronization point of the same process image partition ID was reached. Or the process time for the time-controlled operating modes. • `Int64 in_TimeSinceAnySyncPoint_ns:` The virtual time (in nanoseconds) since the last synchronization point of any process image partition ID was reached. Or the process time for the time-controlled operating modes. • `UInt32 in_SyncPointCount:` The number of synchronization points since the last event. If the events are triggered faster than they are received, multiple events are combined into one event. In this case, this value contains the number of cycles since the last event was received.
Return values	None

Amendment to section 7.8.6.13 SOnSyncPointReachedResult

Description

In contrast to the specifications in the S7-PLCSIM Advanced Function Manual, note the following updated tables:

Table 7 SOnSyncPointReachedResult - Native C++

Syntax	`struct SOnSyncPointReachedResult`
	`{`
	` ERuntimeErrorCode ErrorCode;`
	` SYSTEMTIME SystemTime;`
	` UINT32 Id;`
	` INT64 TimeSinceSameSyncPoint_ns;`
	` INT64 TimeSinceAnySyncPoint_ns;`
	` UINT32 SyncPointCount;`
	`};`

Member	• `ERuntimeErrorCode ErrorCode:` – SREC_TIMEOUT, if no event was triggered during the defined time interval. – `SREC_WARNING_INVALID_CALL`, if no function `RegisterOnSyncPointReachedEvent` was called before. See ERuntimeErrorCode. • `SYSTEMTIME SystemTime:` The virtual system time of the virtual controller at the time when this event was triggered. • `UINT32 Id:` For the `SingleStep_Bus` operating mode: The ID of the IO system that is used for cycle-controlled synchronization of the virtual controller. For all other operating modes: The ID of the process image partition (PIP) that triggers this event. 0 for the cycle control point (End of cycle). • `INT64 TimeSinceSameSyncPoint_ns:` The virtual time (in nanoseconds) since the last synchronization point of the same process image partition ID was reached. For the time-controlled operating modes: The runtime since the last call of the `StartProcessing()` function. • `INT64 TimeSinceAnySyncPoint_ns:` The virtual time (in nanoseconds) since the last synchronization point of any process image partition ID was reached. For the time-controlled operating modes: The runtime since the last call of the `StartProcessing()` function. • `UINT32 SyncPointCount:` The number of synchronization points since the last event. If the events are triggered faster than they are received, multiple events are combined into one event. In this case, this value contains the number of cycles since the last event was received.

Table 8 SOnSyncPointReachedResult - .NET (C#)

Syntax	`struct SOnSyncPointReachedResult`
	`{`
	`ERuntimeErrorCode ErrorCode;`
	`DateTime SystemTime;`
	`UInt32 Id;`
	`Int64 TimeSinceSameSyncPoint_ns;`
	`Int64 TimeSinceAnySyncPoint_ns;`
	`UInt32 SyncPointCount;`
	`}`

Member	• `ERuntimeErrorCode ErrorCode:` – `ERuntimeErrorCode.Timeout`, if no event was triggered during the defined time interval. – `WarningInvalidCall`, if no function `RegisterOnSyncPointReachedEvent` was called before. See ERuntimeErrorCode. • `DateTime DateTime:` The virtual system time of the virtual controller at the time when this event was triggered. • `UInt32 Id:` For the `SingleStep_Bus` operating mode: The ID of the IO system that is used for cycle-controlled synchronization of the virtual controller. For all other operating modes: The ID of the process image partition (PIP) that triggers this event. 0 for the cycle control point (End of cycle). • `Int64 TimeSinceSameSyncPoint_ns:` The virtual time (in nanoseconds) since the last synchronization point of the same process image partition ID was reached. For the time-controlled operating modes: The runtime since the last call of the `StartProcessing()` function. • `Int64 TimeSinceAnySyncPoint_ns:` The virtual time (in nanoseconds) since the last synchronization point of any process image partition ID was reached. For the time-controlled operating modes: The runtime since the last call of the `StartProcessing()` function. • `UInt32 SyncPointCount:` The number of synchronization points since the last event. If the events are triggered faster than they are received, multiple events are combined into one event. In this case, this value contains the number of cycles since the last event was received.

Addition to section 8 Restrictions, messages and solution

Loading the project data of an F-CPU to a standard CPU

When you load project data of an F-CPU to a standard CPU, a window is displayed in the "Load preview" dialog with the query of a password for access to the CPU.

Remedy

1. Cancel the password query with the "Cancel" button.
2. Load the project data again to the standard CPU.

Result: The load operation is performed.

After an upgrade to Windows 10, S7-PLCSIM Advanced no longer starts

After an upgrade to Windows 10 on your PC, S7-PLCSIM Advanced is no longer starting and signals that the virtual Ethernet adapter is missing.

The S7-PLCSIM Advanced Virtual Ethernet adapter was deleted by upgrading to Windows 10.

This behavior also affects the virtual SINUMERIK NCU (Numerical Control Unit) and the SIMIT Simulation Platform.

Remedy

You have the following options after upgrading to Windows 10:

- Repair your S7-PLCSIM Advanced installation.
- Install S7-PLCSIM Advanced once again.

The procedure is described in the S7-PLCSIM Advanced Function Manual.

Error message during simulation start of SIMIT with S7-PLCSIM Advanced

You have configured a S7-PLCSIM Advanced connection in SIMIT. When starting the simulation, SIMIT aborts and the following error message is returned: "`-14 InstanceNotRunning`"

When the error has occurred, you can also no longer open the S7-PLCSIM Advanced instance using the virtual Ethernet adapter.

Remedy

You have the following remedial options:

- Repair your S7-PLCSIM Advanced installation.
- Install S7-PLCSIM Advanced once again.

The procedure is described in the S7-PLCSIM Advanced Function Manual.

Repair or new installation will install the Npcap program library once again. This solves the problem.

Time synchronization via NTP mode

Unlike a real CPU and the settings in the STEP 7 project, the time synchronization of the internal clock of a virtual controller with an NTP server does not work.

T_CONFIG instruction

The instruction T_CONFIG works in S7-PLCSIM Advanced via TCP/IP, but not via softbus.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY