SIEMENS

Desigo DP Transfer Script

siemens.com

General information

Proof of change

Version	Date	Author	Remark
1.0	23.10.2024	Stefan Ott RC-DE SI RDE RPS SAC COS	Document Created

Table of contents

Table of figures

1. Introduction

The data point transfer script is used to write values of source objects to 1..n target objects on a Desigo CC.

The following diagram shows the basic structure of the solution

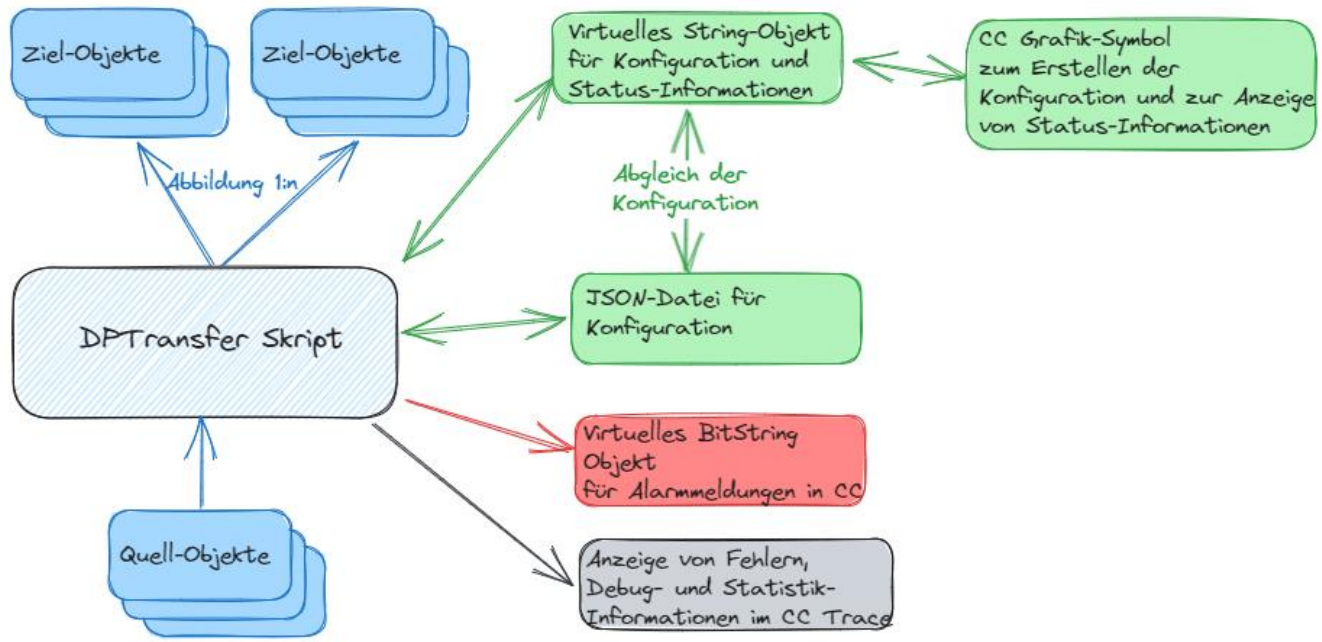

Illustration 1 Schematic structure of the DP Transfer script

2. Description of the solution

Configuration

- The script is configured via a virtual string object and a JSON file
- When the script starts, both configuration sources are compared based on the property CONFIGUPDATETIME
- Both configuration sources are specified as parameters when the script is created
- To create the configuration of the String object, there is a graphic icon in which you can enter the necessary can enter information as a substitution
    - In the graphic icon you can configure a maximum of 5 target objects
    - If more target objects are required, they must be configured in the JSON file
- The JSON file needs to be edited with a file editor

Reading Values

- 1..n source objects are read
- The reading of the source objects takes place cyclically in a 10 s grid (polling) or a change in value is reported (subscription)
- The cycle is controlled by the TRANSFERINTERVAL (0= change in value, all other values are rounded up to the nearest 10 seconds and polled)

Writing Objects

- 1..n target objects can be written per source object
    - For each target object, the target property and the write command must be specified
- Target objects are only written if the value differs from the value of the source object
- Value changes are not transferred to the target objects faster than 10 s
    - If value changes occur in less than 10 s, the status of the transfer goes into 0x01 (delayed COV) and the script remembers that a value transfer is pending and catches up on it (if necessary) in the next polling cycle

- Even if the value does not change, the value of the source object is transferred to the target objects at least every 3600 s (1 h)

Handling

- A source object may only appear once in the entire configuration
  - If there are multiple entries with the same source object, one entry will be displayed when the script is started.
    quenched
    - The choice of which entry to delete depends on the parameter CONFIGUPDATETIME some source object from
      - An entry has a value of "0" -> the object is kept
      - Both entries > 0 and different -> the entry with the larger value is retained
      - Both entries have the same value -> the first entry is preserved
- The script remembers a status for each transfer (source object) and a separate status for each target object
  - Status of the transfer (source object)
    - 0x00 = all OK
      0x01 = pending transfer (COV)
      0x10 = source object unknown
      0x20 = at least one destination object unknown
      0x40 = at least one destination object has write error
      0x80 = configuration has at least one error
  - Status of target objects
    - 0x00 = all OK
      0x10 = write error
      0x20 = object not known
  - If the status of a transfer is 0x10, 0x20, or 0x80, the script is stopped
    - A message is sent via the error object, which must be specified as a parameter when creating the script
  - If the status of a transfer has the value 0x40 and no further error occurs (see above), the script continues to run
    - A message is sent via the error object, which must be specified as a parameter when creating the script
  - If the status of all transfers is 0x00 or 0x01, the script will continue to run and there will be no message

3. Creating a script

To create the script you first have to use the command INCLUDE the script DPSYNC400 from the library GLOBAL_TOOLBOX400 Import into a variable
By calling the method STARTDPTRANSFER400 the script is started.

- When calling the method, you have to specify the following parameters
  - Reference to a virtual string object that contains the configuration for the script. When importing the toolbox, an example object is created.
  - Reference to a virtual BitString object that is necessary for displaying error messages in the CC. When importing the toolbox, an example object is created
  - Reference to a JSON file that contains the configuration. If the file does not exist, it will be created. You can specify a path. This path is relative to
    <PROJEKTVERZEICHNIS>\SHARED\SCRIPTING
- Multiple instances of the script can be created. For each instance, however, you need its own configuration string object, its own error BitString object, and its own JSON file

Example of a call

```js
var DPTransferExample= include("DPSync400", "Global_ToolBox400_1");
DPTransferExample.StartDPTransfer400
                        ("System1.ApplicationView:ApplicationView.Logics.VirtualObjects.DataTransfer400.DataTransferConfiguration",
                         "System1.ApplicationView:ApplicationView.Logics.VirtualObjects.DataTransfer400.DataTransferError",
                         "DPTransfer400/DP_Transfer_config400.json");
```

Illustration 2 Example of creating a script

4. Configuring the script via JSON file

The DPTransfer script is configured via a JSON string, which can be passed either via a virtual Desigo CC string object, or/and a text file.
The string object and the text file must be specified when the script is created. The String object must exist. If the file does not exist, it will be created.
The file is stored in the subdirectory <PROJEKTVERZEICHNIS>\SHARED\SCRIPTING created. You can also define a path for the file. This is relative to the subdirectory mentioned above.
There Paramter DP_TRANSFER/DP_TRANSFER_CONFIG.JSON (Attention: Specify path with forward slash "/"!) e.g. creates the file <PROJEKTVERZEICHNIS>\SHARED\SCRIPTING\DP_TRANSFER\DP_TRANSFER2_CONFIG.JSON
Here's an example of a configuration file with explanations of the parameters. Comments are written in angle brackets "<" and ">"

```json
[  ----- <Die Konfiguration besteht aus einem Array von Transfer-Konfigurationen>
  { ---- < Transfer-Konfigurations Eintrag>
       "configUpdateTime": 1700753979367, ----- < Zeitstempel der letzten Änderung. Beim Start des Skripts werden die Parameter
                                                  des aktuelleren Eintrags (Textdatei oder String-Objekt) übernommen
                                                      Spezielle Zahlen:
                                                          "0" bedeutet, dass dieser Eintrag Priorität hat und beim nächsten Start
                                                              die Parameter im String-Objekt überschrieben werden
                                                          "-1" bedeutet, dass dieser Eintrag beim nächsten Start des Skripts aus der
                                                              Textdatei und dem String-Objekt gelöscht wird >
       "dst": [ ----- <Array von Ziel-Objekt-Konfigurationen>
           { ----- < Ziel-Objekt konfigurationseintrag>
               "dstObj": "System1.ApplicationView:ApplicationView.Logics.VirtualObjects.virtAnalog2;", ----- <Objekt das beschrieben werden
                                                                                                              soll>
               "dstProp": "Value",   ----- <Property, dass beschrieben werden soll>
               "state": 0, ----- <Statusparameter>
               "writeCmd": "Write" ----- < Command das zum Schreiben des Ziel-Objektes verwendet werden soll.
                                            Die möglichen Commands sind vom Objektmodell abhängig.
                     Bevorzugt sollten folgende Commands verwendet werden:
                         - virtuelle Objekte: WriteInternal
                         - Desigo PX: WritePrio08
                         - BACnet 3rd: Write
                         - Modbus: Write
                         Bis auf die virtuellen Objekte schreiben alle o.g. Befehle Werte in die Log-Datenbank.
                         Evtl. kann es sinnvoll sein das Objektmodell projektspezifisch zu vererben und ein Command
                         zu definieren das nicht logt!>
    },
    {
      "dstObj": "System1.ApplicationView:ApplicationView.Logics.VirtualObjects.Dst2;",
      "dstProp": "Value",
      "state": 0,
      "writeCmd": "Write"
    },
    {
      "dstObj": "System1.ManagementView:ManagementView.FieldNetworks.Network.Hardware.TestDevice.BV_5;",
      "dstProp": "Present_Value",
      "state": 0,
      "writeCmd": "WritePrio08"
    },
    {
      "dstObj": "System1.ManagementView:ManagementView.FieldNetworks.Network.Hardware.TestDevice.AV_5;",
      "dstProp": "Present_Value",
      "state": 0,
      "writeCmd": "WritePrio08"
    }
    ], ---- < Ende Array für Zielkonfigurationen>
    "lastTransferTime": 0, ----- < Statusparameter>
    "lastTransferValue": null, ----- < Statusparameter>
    "nextTransferTime": 0, ------ < Statusparameter>
    "srcObj": "System1.ManagementView:ManagementView.FieldNetworks.Network.Hardware.TestDevice.CS_1;", ---- < Quellobjekt>
    "srcProp": "Present_Value", ---- <Property des Quellobjekts das übertragen werden soll>
    "srcValue": null, --- < Statusparameter>
    "state": 0, ----- < Statusparameter>
    "transferInterval": "20" ---- < legt den Pollingintervall in Sekunden fest. Der Wert wird auf die nächsten 10er Sekunden aufgerundet
                    - "0" bedeutet, dass Werte per COV übertragen werden
    } ----- < Ende Transfer-Konfigurations Eintrag. Weitere Einträge werden durch Komma abgetrennt>
    ] ----- < Ende Array der Transfer-Konfigurationen>
    */
```

Illustration 3 Example of a configuration file with explanation of the parameters

5. Configuring the script via icon

The String object can be defined by the DYN_ALL_GENERIC_DISPLAY_DPTRANSFER_CONFIG_400 can be configured.
For details, see the description of the symbol further back.

6. Status Messages

6.1 Status indicators in the icon

DYN_ALL_GENERIC_DISPLAY_DPTRANSFER_CONFIG_400
- Displays status information for a transfer
- The information of the source object and up to 5 target objects is displayed

DYN_ALL_GENERIC_DISPLAY_DPTRANSFER_DISPLAY_400
- Displays the status information of all transfers
- Only the information of the source objects is displayed

6.2 Error messages in CC about BitString virtual object

Error messages are displayed via a virtual bit-string object in the CC.
The object must be referenced when creating the script
When importing the toolbox, a sample object with alarm configuration is created

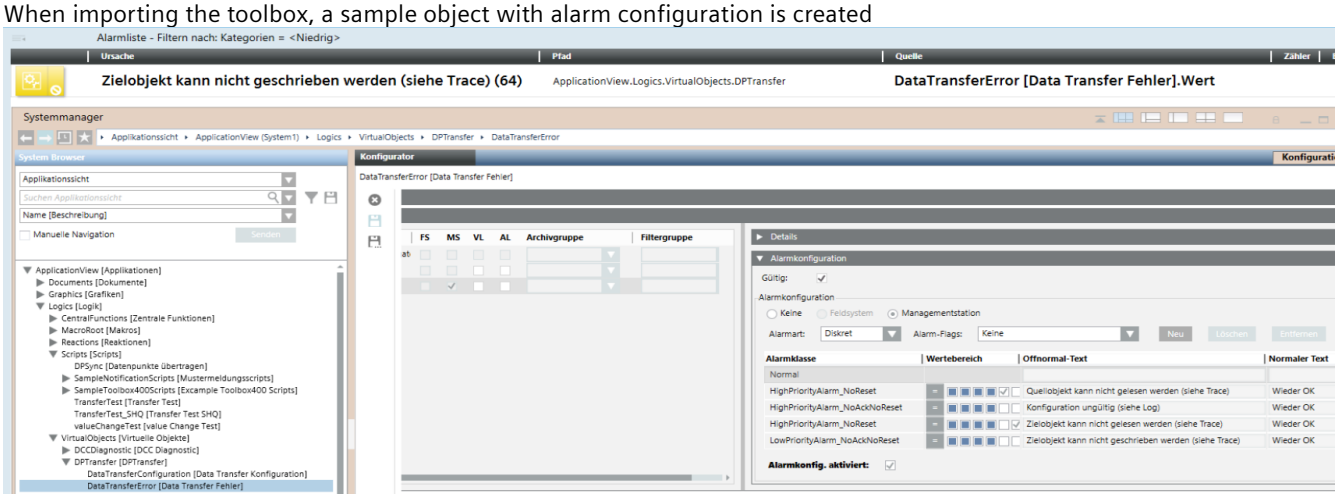

Illustration 4 Alarm Configuration of the BitString Object

7. Trace information via script

In the trace viewer you can turn on a trace channel for the script

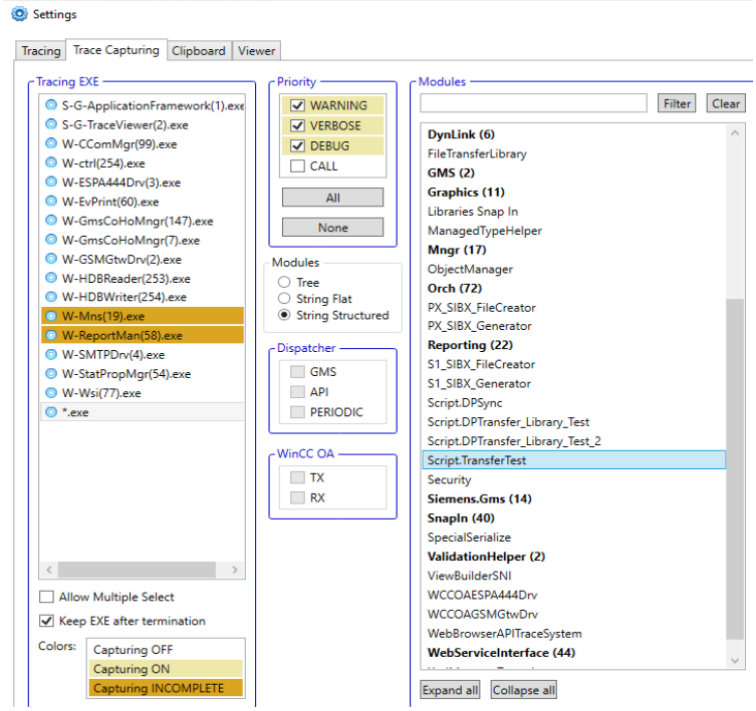

Illustration 5 Trace channel for the DP transfer script in the CC Trace Viewer

You will see debug and statistics information there

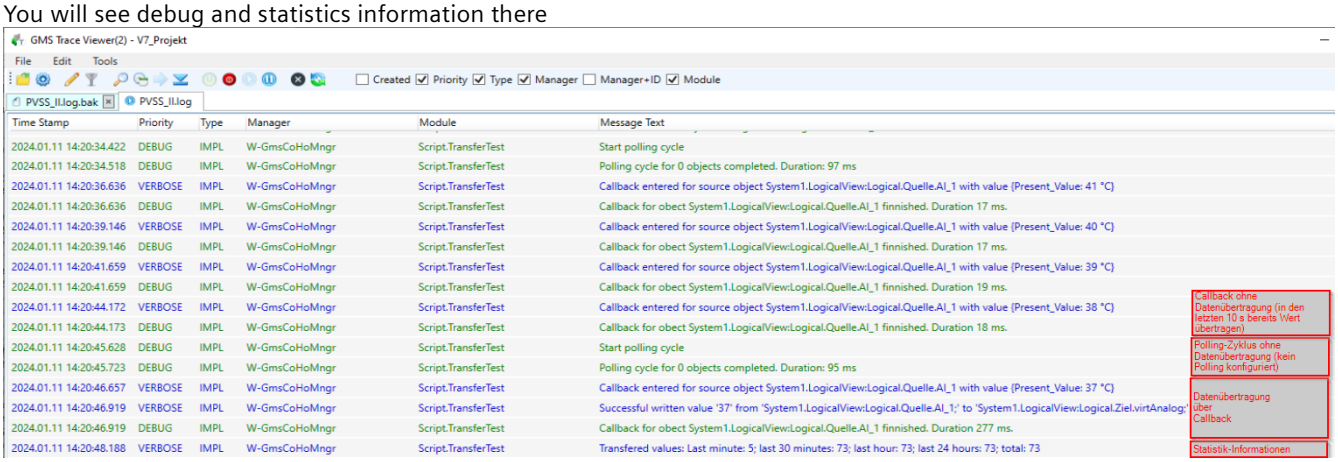

Illustration 6 Information displayed in the CC Trace Viewer via the DP Transfer Script

8. Description of Symbols

8.1 DYN_All_Generic_Display_DPTransfer_Config_400

8.1.1 General

The symbol DYN_All_Generic_Display_DPTransfer_Config_400 is used to create a configuration for the DP Transfer script and displays status information of the script.

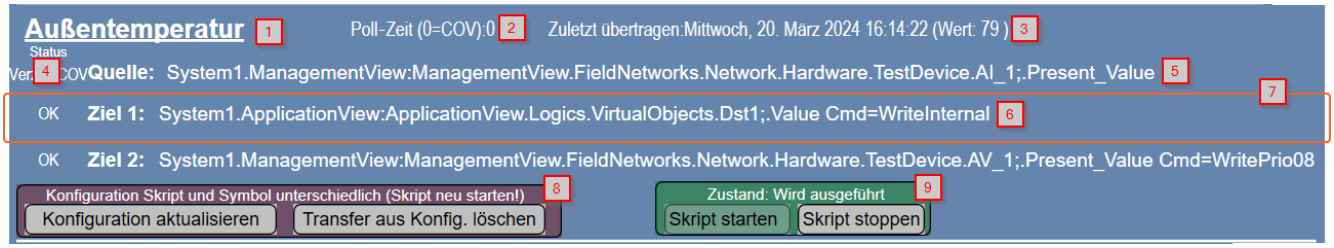

Illustration 7 The symbol DYN_All_Generic_Display_DPTransfer_Config_400

1. Display a configurable title
2. Display of Poll Time
3. Display of the last value transferred and the time when it was transferred
4. Display of a status text for the source and all target objects
5. Displaying the Source Object and the Source Property
6. 1..5 target objects can be displayed. The size of the icon changes dynamically. For each target object, the object path, the target property, and the command used are displayed
7. There is a selection reference for the source and all target objects, as well as the script node and the configuration object
8. Here you can transfer the substitutions entered in the symbol to the script configuration, or delete them from it. It also shows if there is a difference between the configured substititutions and the script configuration. The source object is used to search. A change in the configuration will only be applied after the script has been restarted.
9. Here you can see the state of the script and start or stop it

8.1.2 Substitutions

Title : Enter a title for the transfer. Default value "DP Transfer"
Direction: Adjusts the curves and hyphens of the symbol so that it can be integrated into a list (Standalone, Top, Middle, Bottom)
Width: Width of the icon. Default value: 930
FontSize: Size of the font. Individual texts (e.g. titles) are offset to this substitution
NumberOfDestinations: Number of target objects that are described. Changes the height of the icon. Default value: 1. Value range: 1..5
ScriptReference: Reference to the node that contains the script
The 2013-2016 World No.: Reference to a virtual string object that contains the configuration of the script
srcObj: Object from which the value is read (source), which is then written to the target objects
srcProp: Property of the source object that is being read
Interval: Interval in s in which the source object is read. "0" means change in value. The script rounds the interval to the nearest 10 seconds.
DST1OBJ/ DST1Prop/ DST1CMD DST5OBJ/ DST5prop/ DST5CMD: Configure up to 5 target objects. For each destination, the reference to the object, the property to be written and the command to be used (Write, Comand,...) must be specified.

8.2 DYN_All_Generic_Display_DPTransfer_Display_400

8.2.1 General

The symbol DYN_All_Generic_Display_DPTransfer_Display_400 displays the current configuration of the DP Transfer script with the most important parameters

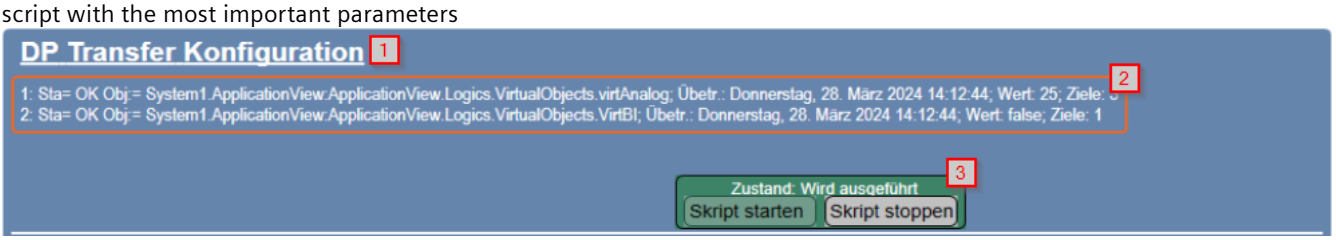

Illustration 8 The symbol DYN_All_Generic_Display_DPTransfer_Display_400

1. Display a configurable title
2. View all configured transfers with the following information
    - Status des Transfers
    - Source object of the transfer
    - Timestamps of when the value of the source object was last transferred
    - Value transferred to the target objects
    - Number of target objects
    - There is a selection reference for the configuration object and the script reference
        - 3. Here you can see the state of the script and start or stop it

8.2.2 Substitutions

Title : Enter a title for the transfer. Default value "DP Transfer Configuration"
Direction: Adjusts the curves and hyphens of the symbol so that it can be integrated into a list (Standalone, Top, Middle, Bottom)
Height: Adjusts the height of the icon
Width: Width of the icon. Default value: 930
FontSize: Size of the font. Individual texts (e.g. titles) are offset to this substitution
ScriptReference: Reference to the node that contains the script
The 2013-2016 World No.: Reference to a virtual string object that contains the configuration of the script