



SIEMENS

Desigo DP Transfer Skript

[siemens.com](https://www.siemens.com)

Allgemeine Informationen

Änderungsnachweis

Version	Datum	Autor	Bemerkung
1.0	23.10.2024	Stefan Ott RC-DE SI RDE RPS SAC COS	Dokument erstellt

Inhaltsverzeichnis

1.	Einleitung	4
2.	Beschreibung der Lösung	4
3.	Skript erstellen	5
4.	Konfigurieren des Skripts über JSON-Datei.....	6
5.	Konfigurieren des Skripts über Symbol	7
6.	Status Meldungen	7
6.1	Statusanzeigen im Symbol	7
6.2	Fehlermeldungen in CC über virtuelles BitString Objekt	7
7.	Trace-Informationen über Skript	8
8.	Beschreibung der Symbole	9
8.1	DYN_All_Generic_Display_DPTransfer_Config_400	9
8.1.1	Allgemein	9
8.1.2	Substitutions	9
8.2	DYN_All_Generic_Display_DPTransfer_Display_400	10
8.2.1	Allgemein	10
8.2.2	Substitutions	10

Abbildungsverzeichnis

Abbildung 1	Schematischer Aufbau des DP Transfer Skriptes.....	4
Abbildung 2	Beispiel für das Erstellen eines Skriptes.....	5
Abbildung 3	Beispiel für eine Konfigurationsdatei mit Erklärung der Parameter	6
Abbildung 4	Alarmkonfiguration des BitString Objekts	7
Abbildung 5	Trace-Kanal für das DP-Transfer Skript im CC Trace Viewer	8
Abbildung 6	Informationen, die im CC Trace-Viewer über das DP-Transfer Skript angezeigt werden	8
Abbildung 7	Das Symbol DYN_All_Generic_Display_DPTransfer_Config_400.....	9
Abbildung 8	Das Symbol DYN_All_Generic_Display_DPTransfer_Display_400	10

1. Einleitung

Das Datenpunkt-Transfer Skript dient dazu auf einer Desigo CC Werte von Quell-Objekten auf 1..n Ziel-Objekte zu schreiben.

Das folgende Schema zeigt den prinzipiellen Aufbau der Lösung

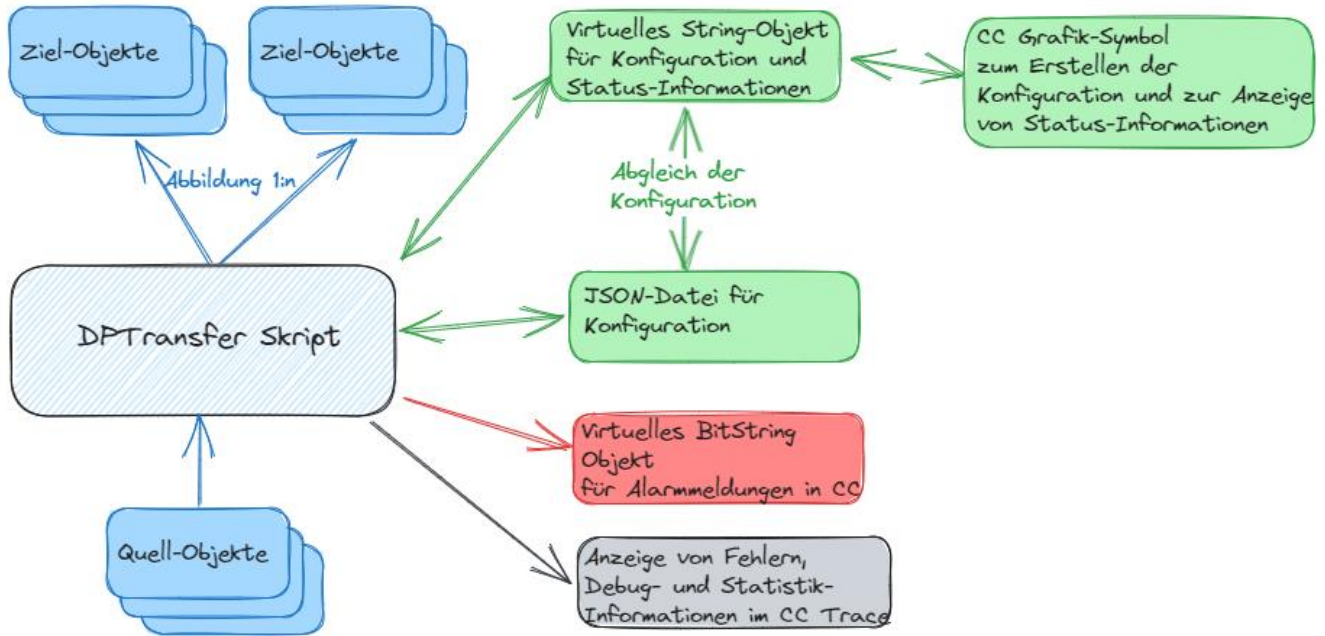


Abbildung 1 Schematischer Aufbau des DP Transfer Skriptes

2. Beschreibung der Lösung

Konfiguration

- Das Skript wird über ein virtuelles String-Objekt und eine JSON-Datei konfiguriert
- Beim Start des Skripts erfolgt ein Abgleich beider Konfigurationsquellen basierend auf der Eigenschaft CONFIGUPDATE TIME
- Beide Konfigurationsquellen werden beim Erstellen des Skripts als Parameter angegeben
- Für das Erstellen der Konfiguration des String-Objektes gibt es ein Grafik-Symbol, in dem man die nötigen Informationen als Substitutions eingeben kann
 - Im Grafik-Symbol kann man maximal 5 Ziel-Objekte konfigurieren
 - Wenn mehr Ziel-Objekte notwendig sind, müssen diese in der JSON-Datei konfiguriert werden
- Die JSON-Datei muss mit einem Datei-Editor bearbeitet werden

Lesen von Werten

- Es werden 1..n Quell-Objekte gelesen
- Das Lesen der Quell-Objekte geschieht zyklisch im 10 s Raster (Polling) oder es wird eine Wertänderung gemeldet (Abonnement)
- Der Zyklus wird über die Eigenschaft TRANSFERINTERVAL gesteuert (0= Wertänderung, alle anderen Werte werden auf die nächsten 10er Sekunden aufgerundet und gepollt)

Schreiben von Objekten

- Pro Quell-Objekt können 1..n Ziel-Objekte geschrieben werden
 - Für jedes Ziel-Objekt muss das Ziel-Property und das Schreib-Command angegeben werden
- Ziel-Objekte werden nur geschrieben, wenn sich der Wert vom Wert des Quell-Objektes unterscheidet
- Wertänderungen werden nicht schneller als 10 s auf die Ziel-Objekte übertragen
 - Treffen Wertänderungen in weniger als 10 s ein, so geht der Status des Transfers in 0x01 (verzögertes COV) und das Skript merkt sich, dass eine Wertübertragung aussteht und holt diese (falls nötig) beim nächsten Polling-Zyklus nach

- Auch wenn sich der Wert nicht ändert, wird spätestens alle 3600 s (1 h) der Wert des Quell-Objektes auf die Ziel-Objekte übertragen

Fehlerbehandlung

- Ein Quell-Objekt darf in der gesamten Konfiguration nur einmal vorkommen
 - Wenn es mehrere Einträge mit demselben Quell-Objekt gibt wird ein Eintrag beim Start des Skript gelöscht
 - Die Auswahl, welcher Eintrag gelöscht wird, hängt vom Parameter CONFIGUPDATETIME des Quell-Objektes ab
 - Ein Eintrag hat den Wert "0" -> das Objekt wird behalten
 - Beide Einträge > 0 und unterschiedlich -> der Eintrag mit dem größeren Wert bleibt erhalten
 - Beide Einträge haben denselben Wert -> der erste Eintrag bleibt erhalten
- Das Skript merkt sich für jeden Transfer (Quell-Objekt) einen Status und für jedes Ziel-Objekt einen separaten Status
 - Stati des Transfers (Quell-Objekt)
 - 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
 - Stati der Ziel-Objekte
 - 0x00 = all OK
 - 0x10 = write error
 - 0x20 = object not known
 - Wenn der Status eines Transfers den Wert 0x10, 0x20 oder 0x80 hat, wird das Skript gestoppt
 - Eine Meldung erfolgt über das Fehlerobjekt das beim Erstellen des Skripts als Parameter mit angegeben werden muss
 - Wenn der Status eines Transfers den Wert 0x40 hat und kein weiterer Fehler auftritt (s.o.), läuft das Skript weiter
 - Eine Meldung erfolgt über das Fehlerobjekt das beim Erstellen des Skripts als Parameter mit angegeben werden muss
 - Wenn der Status aller Transfers den Wert 0x00 oder 0x01 hat , läuft das Skript weiter und es gibt keine Meldung

3. Skript erstellen

Zum Erstellen des Skripts muss man zuerst über den Befehl INCLUDE das Skript DPSYNC400 aus der Bibliothek GLOBAL_TOOLBOX400 in eine Variable importieren

Durch Aufrufen der Methode STARTDPTTRANSFER400 wird das Skript gestartet.

- Beim Aufruf der Methode muss man folgende Parameter mit angeben
 - Referenz auf ein virtuelles String-Objekt das die Konfiguration für das Skript enthält. Beim Import der Toolbox wird ein Beispielobjekt erzeugt.
 - Referenz auf ein virtuelles BitString-Objekt das für die Anzeige von Fehlermeldungen in der CC nötig ist. Beim Import der Toolbox wird ein Beispielobjekt erzeugt
 - Referenz auf eine JSON-Datei, die die Konfiguration enthält. Wenn die Datei nicht existiert, wird sie erzeugt. Man kann einen Pfad mit angeben. Dieser Pfad ist relativ zu <PROJEKTVERZEICHNIS>\SHARED\SCRIPTING
- Vom Skript können mehrere Instanzen erstellt werden. Für jede Instanz braucht man dann aber ein eigenes Konfigurations-String-Objekt, ein eigenes Fehler-BitString-Objekt und eine eigene JSON-Datei

Beispiel für einen Aufruf

```

1 var DPTransferExample= include("DPSync400", "Global_ToolBox400_1");
2 DPTransferExample.StartDPTTransfer400
3     ("System1.ApplicationView:ApplicationView.Logics.VirtualObjects.DataTransfer400.DataTransferConfiguration",
4     "System1.ApplicationView:ApplicationView.Logics.VirtualObjects.DataTransfer400.DataTransferError",
5     "DPTransfer400/DP_Transfer_config400.json");

```

Abbildung 2 Beispiel für das Erstellen eines Skriptes

4. Konfigurieren des Skripts über JSON-Datei

Das DPTransfer Skript wird über einen JSON-String konfiguriert, die entweder über ein virtuelles Desigo CC String-Objekt, oder/und eine Textdatei übergeben werden kann.

Das String-Objekt und die Textdatei müssen beim Erstellen des Skripts mit angegeben werden. Das String-Objekt muss existieren. Wenn die Datei nicht existiert, wird sie angelegt.

Die Datei wird im Unterverzeichnis <PROJEKTVERZEICHNIS>\SHARED\SCRIPTING erstellt. Man kann für die Datei auch einen Pfad definieren. Dieser ist relativ zum vorn genannten Unterverzeichnis.

Der Parameter DP_TRANSFER/DP_TRANSFER_CONFIG.JSON (Achtung: Pfad mit vorwärts-Schrägstrich "/" angeben!) erstellt z.B. die Datei <PROJEKTVERZEICHNIS>\SHARED\SCRIPTING\DP_TRANSFER\DP_TRANSFER2_CONFIG.JSON

Hier ein Beispiel für eine Konfigurationsdatei mit Erläuterungen zu den Parametern. Kommentare stehen in spitzen Klammern "<" und ">"

```
1 [ ----- <Die Konfiguration besteht aus einem Array von Transfer-Konfigurationen>
2 { ----- < Transfer-Konfigurations Eintrag>
3   "configUpdateTime": 1700753979367, ----- < Zeitstempel der letzten Änderung. Beim Start des Skripts werden die Parameter
4     des aktuelleren Eintrags (Textdatei oder String-Objekt) übernommen
5     Spezielle Zahlen:
6       "0" bedeutet, dass dieser Eintrag Priorität hat und beim nächsten Start
7       die Parameter im String-Objekt überschrieben werden
8       "-1" bedeutet, dass dieser Eintrag beim nächsten Start des Skripts aus der
9       Textdatei und dem String-Objekt gelöscht wird >
10  "dst": [ ----- <Array von Ziel-Objekt-Konfigurationen>
11    { ----- < Ziel-Objekt Konfigurationseintrag>
12      "dstObj": "System1.ApplicationView:ApplicationView.Logics.VirtualObjects.virtAnalog2;", ----- <Objekt das beschrieben werden
13        soll>
14      "dstProp": "Value", ----- <Property, dass beschrieben werden soll>
15      "state": 0, ----- <Statusparameter>
16      "writeCmd": "Write" ----- < Command das zum Schreiben des Ziel-Objektes verwendet werden soll.
17        Die möglichen Commands sind vom Objektmodell abhängig.
18        Bevorzugt sollten folgende Commands verwendet werden:
19          - virtuelle Objekte: WriteInternal
20          - Desigo PX: WritePrio08
21          - BACnet 3rd: Write
22          - Modbus: Write
23        Bis auf die virtuellen Objekte schreiben alle o.g. Befehle Werte in die Log-Datenbank.
24        Evtl. kann es sinnvoll sein das Objektmodell projektspezifisch zu vererben und ein Command
25        zu definieren das nicht logt!>
26    },
27    {
28      "dstObj": "System1.ApplicationView:ApplicationView.Logics.VirtualObjects.Dst2;",
29      "dstProp": "Value",
30      "state": 0,
31      "writeCmd": "Write"
32    },
33    {
34      "dstObj": "System1.ManagementView:ManagementView.FieldNetworks.Network.Hardware.TestDevice.BV_5;",
35      "dstProp": "Present_Value",
36      "state": 0,
37      "writeCmd": "WritePrio08"
38    },
39    {
40      "dstObj": "System1.ManagementView:ManagementView.FieldNetworks.Network.Hardware.TestDevice.AV_5;",
41      "dstProp": "Present_Value",
42      "state": 0,
43      "writeCmd": "WritePrio08"
44    }
45  ], ----- < Ende Array für Zielkonfigurationen>
46  "lastTransferTime": 0, ----- < Statusparameter>
47  "lastTransferValue": null, ----- < Statusparameter>
48  "nextTransferTime": 0, ----- < Statusparameter>
49  "srcObj": "System1.ManagementView:ManagementView.FieldNetworks.Network.Hardware.TestDevice.CS_1;", ----- < Quellobjekt>
50  "srcProp": "Present_Value", ----- <Property des Quellobjekts das übertragen werden soll>
51  "srcValue": null, --- < Statusparameter>
52  "state": 0, ----- < Statusparameter>
53  "transferInterval": "20" ----- < legt den Pollingintervall in Sekunden fest. Der Wert wird auf die nächsten 10er Sekunden aufgerundet
54    - "0" bedeutet, dass Werte per COV übertragen werden
55  } ----- < Ende Transfer-Konfigurations Eintrag. Weitere Einträge werden durch Komma abgetrennt>
56 ] ----- < Ende Array der Transfer-Konfigurationen>
57 */
58
```

Abbildung 3 Beispiel für eine Konfigurationsdatei mit Erklärung der Parameter

5. Konfigurieren des Skripts über Symbol

Das String-Objekt kann über das Symbol DYN_ALL_GENERIC_DISPLAY_DPTRANSFER_CONFIG_400 konfiguriert werden. Für Details siehe Beschreibung des Symbols weiter hinten.

6. Status Meldungen

6.1 Statusanzeigen im Symbol

DYN_ALL_GENERIC_DISPLAY_DPTRANSFER_CONFIG_400

- Zeigt Status-Informationen für einen Transfer an
- Es werden die Informationen des Quell-Objekts und von bis zu 5 Ziel-Objekten angezeigt

DYN_ALL_GENERIC_DISPLAY_DPTRANSFER_DISPLAY_400

- Zeigt die Status-Informationen von allen Transfers an
- Es werden nur die Informationen der Quell-Objekte angezeigt

6.2 Fehlermeldungen in CC über virtuelles BitString Objekt

Fehlermeldungen werden über ein virtuelles Bit-String Objekt in der CC angezeigt.

Das Objekt muss beim Erstellen des Skripts referenziert werden

Beim Import der Toolbox wird ein Beispiel-Objekt mit Alarmkonfiguration erstellt

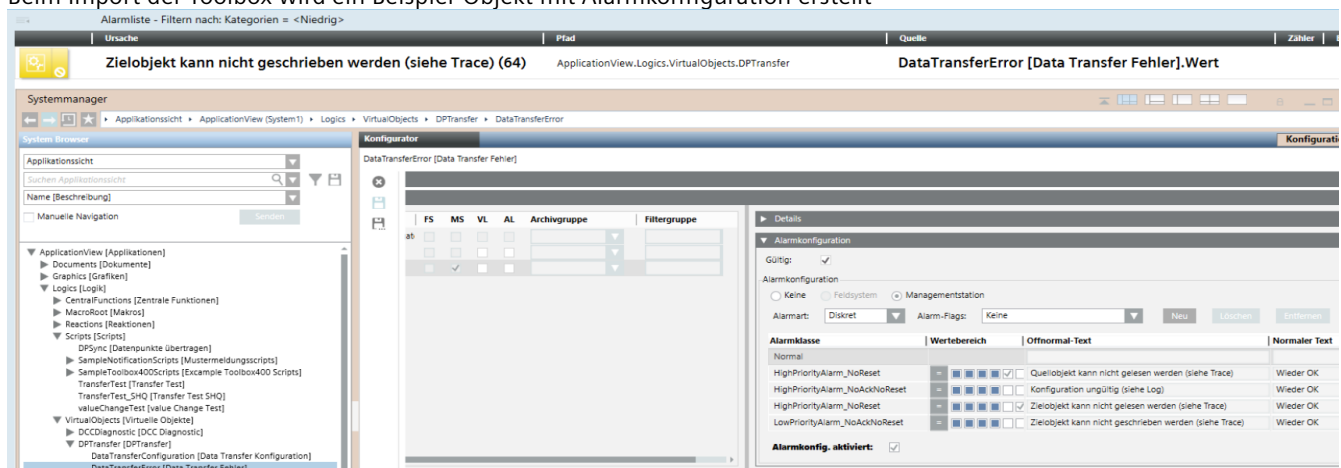


Abbildung 4 Alarmkonfiguration des BitString Objekts

7. Trace-Informationen über Skript

Im Trace-Viewer kann man für das Skript einen Trace-Kanal einschalten

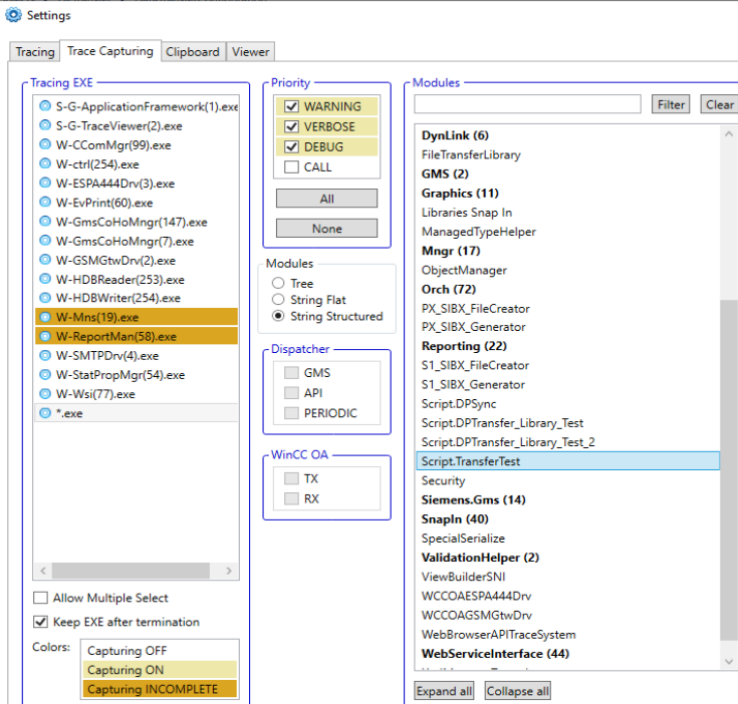


Abbildung 5 Trace-Kanal für das DP-Transfer Skript im CC Trace Viewer

Man sieht dort dann Debug- und Statistik-Informationen

Time Stamp	Priority	Type	Manager	Module	Message Text
2024.01.11 14:20:34.422	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Start polling cycle
2024.01.11 14:20:34.518	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Polling cycle for 0 objects completed. Duration: 97 ms
2024.01.11 14:20:36.636	VERBOSE	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback entered for source object System1.LogicalView.Logical.Quelle.AL_1 with value (Present_Value: 41 °C)
2024.01.11 14:20:36.636	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback for object System1.LogicalView.Logical.Quelle.AL_1 finished. Duration 17 ms.
2024.01.11 14:20:39.146	VERBOSE	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback entered for source object System1.LogicalView.Logical.Quelle.AL_1 with value (Present_Value: 40 °C)
2024.01.11 14:20:39.146	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback for object System1.LogicalView.Logical.Quelle.AL_1 finished. Duration 17 ms.
2024.01.11 14:20:41.659	VERBOSE	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback entered for source object System1.LogicalView.Logical.Quelle.AL_1 with value (Present_Value: 39 °C)
2024.01.11 14:20:41.659	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback for object System1.LogicalView.Logical.Quelle.AL_1 finished. Duration 19 ms.
2024.01.11 14:20:44.172	VERBOSE	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback entered for source object System1.LogicalView.Logical.Quelle.AL_1 with value (Present_Value: 38 °C)
2024.01.11 14:20:44.173	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback for object System1.LogicalView.Logical.Quelle.AL_1 finished. Duration 18 ms.
2024.01.11 14:20:45.628	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Start polling cycle
2024.01.11 14:20:45.723	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Polling cycle for 0 objects completed. Duration: 95 ms
2024.01.11 14:20:46.657	VERBOSE	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback entered for source object System1.LogicalView.Logical.Quelle.AL_1 with value (Present_Value: 37 °C)
2024.01.11 14:20:46.919	VERBOSE	IMPL	W-GmsCoHoMngr	Script.TransferTest	Successful written value '37' from 'System1.LogicalView.Logical.Quelle.AL_1;' to 'System1.LogicalView.Logical.Ziel.virtAnalog'
2024.01.11 14:20:46.919	DEBUG	IMPL	W-GmsCoHoMngr	Script.TransferTest	Callback for object System1.LogicalView.Logical.Quelle.AL_1 finished. Duration 277 ms.
2024.01.11 14:20:48.188	VERBOSE	IMPL	W-GmsCoHoMngr	Script.TransferTest	Transferred values: Last minute: 5; last 30 minutes: 73; last hour: 73; last 24 hours: 73; total: 73

Abbildung 6 Informationen, die im CC Trace-Viewer über das DP-Transfer Skript angezeigt werden

8. Beschreibung der Symbole

8.1 DYN_All_Generic_Display_DPTransfer_Config_400

8.1.1 Allgemein

Das Symbol DYN_ALL_GENERIC_DISPLAY_DPTRANSFER_CONFIG_400 dient dazu eine Konfiguration für das DP Transfer Skript zu erstellen und zeigt Status-Informationen des Skripts an.



Abbildung 7 Das Symbol DYN_All_Generic_Display_DPTransfer_Config_400

1. Anzeige eines konfigurierbaren Titels
2. Anzeige der Poll-Zeit
3. Anzeige, des zuletzt übertragenen Wertes und des Zeitpunkts, wann dieser übertragen wurde
4. Anzeige eines Status-Textes für das Quell- und alle Ziel-Objekte
5. Anzeige des Quell-Objekts und des Quell-Properties
6. Es können 1..5 Ziel-Objekte angezeigt werden. Die Größe des Symbols ändert sich dynamisch. Für jedes Ziel-Objekt wird der Objekt Pfad, das Ziel-Property und das verwendete Command angezeigt
7. Für das Quell- und alle Ziel-Objekte, sowie den Skript-Knoten und das Konfigurationsobjekt gibt es eine Auswahlreferenz
8. Hier kann man die im Symbol eingetragenen Substitutions in die Skript-Konfiguration übernehmen, oder sie daraus löschen. Hier wird auch angezeigt, wenn es einen Unterschied zwischen den konfigurierten Substitutions und der Skript-Konfiguration gibt. Zum Suchen wird das Quell-Objekt verwendet. Eine Änderung der Konfiguration wird erst nach dem Neustart des Skripts übernommen.
9. Hier sieht man den Zustand des Skripts und kann es starten oder stoppen

8.1.2 Substitutions

TITLE : Eingabe eines Titels für den Transfer. Standardwert "DP Transfer"

DIRECTION: Passt die Rundungen und die Trennstiche des Symbols so an, dass man es in eine Liste integrieren kann (Alleinstehend, Oben, Mitte, Unten)

WIDTH: Breite des Symbols. Standardwert: 930

FONTSIZE: Größe der Schrift. Einzelne Texte (z.B. Title) sind mit einem Offset auf diese Substitution versehen

NUMBEROFDESTINATIONS: Anzahl der Zielobjekte, die beschrieben werden. Verändert die Höhe des Symbols. Standardwert: 1. Wertebereich: 1..5

SCRIPTREFERENCE: Referenz auf den Knoten, der das Skript enthält

TRANSFERCONFIGOBJ: Referenz auf ein virtuelles String-Objekt, das die Konfiguration des Skripts enthält

SRCOBJ: Objekt von dem der Wert gelesen wird (Quelle), der dann auf die Ziel-Objekte geschrieben wird

SRCPROP: Property des Quell-Objekts, das gelesen wird

INTERVAL: Intervall in s in dem das Quell-Objekt gelesen wird. "0" bedeutet Wertänderung. Der Intervall wird vom Skript auf die nächsten 10er Sekunden gerundet.

DST1OBJ/ DST1PROP/ DST1CMD ... DST5OBJ/ DST5PROP/ DST5CMD: Konfigurieren von bis zu 5 Ziel-Objekten. Pro Ziel muss die Referenz aufs Objekt, das zu schreibende Property und das zu verwendende Kommando (Write, Comand,...) angegeben werden.

8.2 DYN_All_Generic_Display_DPTransfer_Display_400

8.2.1 Allgemein

Das Symbol DYN_ALL_GENERIC_DISPLAY_DPTRANSFER_DISPLAY_400 zeigt die aktuelle Konfiguration des DP Transfer Skripts mit den wichtigsten Parametern an

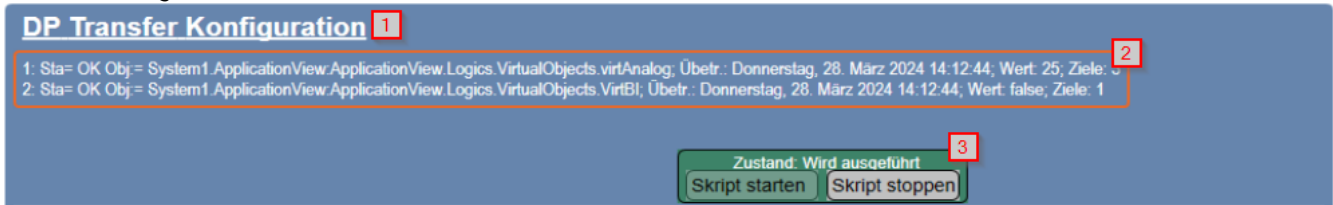


Abbildung 8 Das Symbol DYN_All_Generic_Display_DPTransfer_Display_400

1. Anzeige eines konfigurierbaren Titels
2. Anzeige aller konfigurierten Transfers mit folgenden Informationen
 - Status des Transfers
 - Quell-Objekt des Transfers
 - Zeitstempel, wann der Wert des Quell-Objektes zuletzt übertragen wurde
 - Wert der zu den Ziel-Objekten übertragen wurde
 - Anzahl der Ziel-Objekte
 - Für das Konfigurations-Objekt und die Skript-Referenz gibt es eine Auswahlreferenz
 - 3. Hier sieht man den Zustand des Skripts und kann es starten oder stoppen

8.2.2 Substitutions

TITLE : Eingabe eines Titels für den Transfer. Standardwert "DP Transfer Konfiguration"

DIRECTION: Passt die Rundungen und die Trennstriche des Symbols so an, dass man es in eine Liste integrieren kann (Alleinstehend, Oben, Mitte, Unten)

HEIGHT: Passt die Höhe des Symbols an

WIDTH: Breite des Symbols. Standardwert: 930

FONTSIZE: Größe der Schrift. Einzelne Texte (z.B. Title) sind mit einem Offset auf diese Substitution versehen

SCRIPTREFERENCE: Referenz auf den Knoten, der das Skript enthält

TRANSFERCONFIGOBJ: Referenz auf ein virtuelles String-Objekt, das die Konfiguration des Skripts enthält