Interface Specification

DTP CSharp Help_Manual V 1.0.0.0

Author : Liu Guobin （Manual provided by FFBS.）

VISAS / VERSIONS

	REDACTION	VERIFICATION
Original (V0)	Guobin Liu	Guobin Liu
V1 version		
V2 version		
V3 version		
V4 version		
V5 version		
V6 version		

REVISION	MODIFIED PAGE(S)	DETAILS OF MODIFICATION	DATE
V1	All	Creation	15/10/2019

1 Introduction

1.1 Purpose

This document describes the handling of the existing DTP actions:

dtp_check_and_read

dtp_get_variable

dtp_get_variables

to optimize call performance, calling convention and parameter transfer.

At first run the action "dtp_check_and_read" loads the ATF contents into the internal memory. This contents will be hold durable up to calling this action a second time with parameter "force_to_read" = 1.

A calling of the actions "dtp_get_variable(s)" are possible always at every test run to "refresh" the symbols in the program tables.

It has to guarantee, that the action "dtp_check_and_read" with parameter "force_to_read" = 0 will be call at every test run with the version was got from first run. This handling checks that the currently released version is the same like the loaded version at first run.

With force_to_read = 0 no ATF variables are reload to memory and only the SZ, MLFB and Version is checked if it is RELEASED or READY_TO_CHECK as requested.

If force_to_read = 0 the requested SZ, MLFB and Version are compared against the last call with force_to_read = 1 and if one of them are distinct dtp_check_and_read returns with the error WRONG_PARAMETER_FORCE_TO_READ. The check is added to be safe in the case if the calls with force_to_read = 0 are done with an empty Version string and/or if the SZ,MLFB do change and a call with force_to_read = 1 was missed.

1.2 Abbreviations List

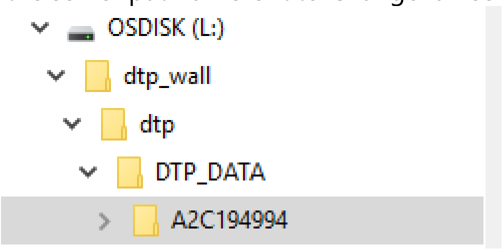
AOI	automatic optical inspection
ATF	address table file
BGA	ball grid array
BIB	boot information block
BIST	built in self test
BOM	bill of material
CAN	controller area network
CC	critical characteristics
CFT DL	cross function testing design and launch
DAE	pressure compensation unit
Div.Q	division quality responsible
DMC	data matrix code
DTP	data to production
DUT	device under test
EE SW	Electrical engineer SW (software developer)
ECR	engineering change request

ECU	electronic control unit
EE	electrical engineer (hardware developer)
EOL	end of line
ESD	electrostatic discharge
HT	high temperature
IC	integrated circuit
ICT	in circuit test
IE test	industrial engineer testing (local test specialist)
IO	information management and organization
IPG	intelligent peripheral interface
IPGTC	intelligent peripheral interface total controller
K-line	communication line
Loc.Q	location quality responsible (manufacturing quality)
LT	low temperature
MD	mechanical design
MM	Manufacturing mode
MOI	manual optical inspection
NPLM	New product launch manager (manufacturing planner)
NVM	non volatile memory
PCB	printed circuit board
PLL	phase-locked loop
PT	process technology
PTS	production test specification
PTS O	PTS owner (EE or TE L)
QMP	quality manager project (quality response)
QMPP	quality manager project production (local Q)
RT	room temperature
SC	special characteristics
SPC	statistical process control
SW	software
TPL	technical project leader
TE E	Test engineer equipment (from CFT DL prototyping department)
TE L	Test engineer product launch (from CFT DL test coverage)
WFC	Wheel Fitted Component
WR	Warranty return center

2 Interface and Functionality

2.1 Dtp_SetDtpServerPath[optional]

This function will be needed if the server path different to Changchun server.



OSDISK (L:)
dtp_wall
dtp
DTP_DATA
A2C194994

In this case Parameter < *sDtpPath* > has to be set to <\\Server x\dtp_wall\dtp\DTP_DATA\>. The folder <0000000000> and <0000000001> are in place of the "Sachzeichnungsnummer (drawing number)" and will be append by the accordingly functions automatically.

Declaration
public **bool** **Dtp_SetDtpServerPath**(**string** *sDtpPath*, out **string** *sError*)

Summary:

Dtp_SetDtpServerPath This function will be needed if the server path different to Changchun server.

PARAMETERS	DESCRIPTION
sDtpPath	*The syntax of path is "\\server\folder\...* *Remark: The string have to terminated with "\"*
sError	*Error information*
RETURN RESULT	**DESCRIPTION**
true	*function has well done; all other codes specify an error.*
false	*Server or path not exist*

2.2 DTP_mlfb_sz_select

This function is optional and give the operator the possibility to select the "Sachzeichnungsnummer"(drawing number) and Mlfb via a dialog box.

At first this function shows a dialog box with the referenced "Sachzeichnungsnummern" (drawing number). The operator can select one of this which will need. After this the function search the MlfB of the given "Sachzeichnungsnummer" depends on Ready_to_check flag and ProcessStep and shows the found MlfB in a second dialog box. The operator can select the MlfB he want and the values are set to the out parameter of the function. This function call is optional and is only used if the test program should be interact manually with the operator.

Declaration
public **int** **DTP_mlfb_sz_select**(**int** *iReady_to_check*, **string** *sSachzeichnungsnummer*,
string *sProcessStep*, out **string** *sSelectedSachzeichnungsnummer*,
out **string** *sSelectedMlfb*, out **string** *sSelectedVersion*, out **string** *sError*)

Summary:

DTP_mlfb_sz_select

PARAMETERS	DESCRIPTION
iReady_to_check	*if ready_to_check=0 only released version the action reads if ready_to_check=1 ready_to_check and released version the action reads if ready_to_check=2 only ready_to_check versions the action reads*
sarrSachzeichnungsnummer	*Parameter Sachzeichnungsnummer to search in dtp-directory.Use wildcards(?) for every position for unknown or different Sachzeichnungsnummer example: 1. Sachzeichnungsnummer 0000012345 2. Sachzeichnungsnummer 0000067891 Param Sachzeichnungsnummer 00000?????*
sProcessStep	*Process Step(e.g.EP; ICT; SPR-OUT.....)*
sSelectedSachzeichnungsnummer	*Returns the selected "Sachzeichnungsnummer" from the dialog*
sSelectedMlfb	*Parameter returns the MlfB of selected ProcessStep*
sSelectedVersion	*Parameter returns the version in relation to Mlfb of the selected ProcessStep*
sError	*Error information*
RETURN RESULT	**DESCRIPTION**
1	*OK*
100	*Directory Sachzeichnungsnummer is empty*
110	*State directory is empty*
180	*Wrong elements in Listbox*
190	*No elements in Listbox selected*

2.3 Dtp_check_and_read

This function checks and (or) reads all contents of existing atf files in the referenced directory. Additional the function can copy files of the subfolder "/data" to the local drive to decrease the access time of binary objects.

Declaration

public int **Dtp_check_and_read**(int *iReady_to_check*, string *sSachzeichnungsnummer*,
string *sMlfb*, string *sProcessStep*, string *sVersion*,
int *iForce_to_read*, int *iCopyFlag*, string *sLocalDriveName*,
out string *sData_Path*, out int *iN_Variables*, out string *sError*)

Summary:

Dtp_check_and_read The action checks or reads depends on parameter "Force_to_read" existing Atf-path / files depends on READY_TO_CHECK Flag detailed informations see description of parameters !! the syntax of path is case sensitiv!! use only for parameterSachzeichnungsnummer; Mlfb;

PARAMETERS	DESCRIPTION
iReady_to_check	*if iReady_to_check = 0 ->The only once released version will read if exist.In this case the parameter strVersion have to initialize to an empty string "" by the caller.The function returns if exist the version of the released atf files* *if iReady_to_check = 1 ->The state of the atf files can be "ready to check" or "released". In this case the parameter strVersion have to filled by the caller depends on the version you want to read.*
sSachzeichnungsnummer	*This parameter is referenced to a valid "sSachzeichnungsnummer" in the dtp path.*
sMlfb	*This parameter is referenced to a valid "sMlfb" in the dtp path.*
sSelectedSachzeichnungsnummer	*Returns the selected "Sachzeichnungsnummer" from the dialog*
sSelectedMlfb	*Parameter returns the MlfB of selected ProcessStep*
sProcessStep	*This parameter is referenced to a valid "ProcessStep" depends on the state of ATF file.*
sVersion	*if iReady_to_check = 0 ->only read RELEASED version will read.* *1. The parameter "sVersion" is empty and the only one released version is read and returns in this parameter.* *2. The parameter "sVersion" is not empty, the action search this referenced version.If the version not released, the action return fail.*
iForce_to_read	*if iForce_to_read = 0 ->only check Atf-path and -files* *if iForce_to_read = 1 ->read all variables of all Atf-files into memory*

sData_Path	*if OK = 1 ->returns in sData_Path the Atf-path for example \\server:\dtp_wall\dtp\Data_Path\Sachzeichnungsnummer>\Mlfb\Version\*
	if OK != 1 ->returns errordescription

iCopyFlag	*This parameter is necassary only to copy files of the subfolder "atf\data" to the local drive.* *iCopyFlag = 0 -> no copy* *iCopyFlag = 1 -> copy if necassary* *iCopyFlag = 2 -> copy and overwrite* *iCopyFlag = 3 -> copy and verify all local files*
iN_Variables	*Output Only: Returns the number of ATF variables read into memory* *if iForce_to_read = 1 else iN_Variables = 0.* *Parameter added in version v3_03 of dtp action dll.*
sLocalDriveName	*This parameter defines the local drive letter by the caller and returns the local path where the files copy to. This parameter is an input- and output-parameter. input: This parameter is necassary only to copy files of the subfolder "atf\data" to the local drive and will be ignore if parameter CopyFlag = 0. The parameter is only the drive letter without ":\" and path. The path, if not exist, creates the action and is the same as DTP-*

	path. output: if the action successful, it returns the full local DTP-path. e.g. C:\dtp\dtp_wall\dtp\dtp_data\ sachzeichnungsnummer \.....
sError	*Description of error if occurred*
RETURN RESULT	**DESCRIPTION**
1	*OK*
-100	*open state file*
-102	*read state file*
-104	*RELEASED version not found*
-106	*i/ o error*
-108	*READY_TO_CHECK not found*
-110	*wrong password*
-112	*ATF directory empty*
-111	*ForceToRead*
－116	*open ATF file*
-210	*Param LocalDriveName not correct*
-220	*create local path to copy*
-230	*check state file of locked versions*
-300	*get local file info*
-310	*get dtp file info*
-320	*get new local file info*
-400	*not enought local disc space*
-410	*erease locked files*
-510	*Verify: open dtp file exception*
-520	*Verify: open local file exception*
-530	*Verify: alloc buffer*
-540	*Verify:read dtp file exception*
-550	*Verify:read local file exception*
-560	*Verify:different size*
-570	*Verify:different contents*
-580	*Verify:local file not found -600*

2.4 Dtp_Get_Variable

Declaration
public **bool** **Dtp_Get_Variable**(**string** *variableName*, out **string** *variableText*, out **string** *sError*)

Summary:

Dtp_Get_Variable

PARAMETERS	DESCRIPTION
variableName	*Variable Name*
variableText	*Variable Text*
sError	*Error information*
RETURN RESULT	**DESCRIPTION**
true	*OK*
false	*ATF does't contain this variable*

Declaration
public **bool** **Dtp_Get_Variables**(**string[]** *variableName*, out **string[]** *variableText*, out **string** *sError*)

Summary:
Dtp_Get_Variables

PARAMETERS	DESCRIPTION
variableName	*Variable Name*
variableText	*Variable Text*
sError	*Error information*
RETURN RESULT	**DESCRIPTION**
true	*OK*
false	*ATF does't contain this variable*

Declaration
public **bool** **Dtp_Get_Variables**(**System.Collections.Generic.List<string>** *variableName*,
out **System.Collections.Generic.List<string>** *variableText*, out **string** *sError*)

Summary:

Dtp_Get_Variables

PARAMETERS	DESCRIPTION
variableName	*Variable Name*
variableText	*Variable Text*

sError	*Error information*
RETURN RESULT	**DESCRIPTION**
true	*OK*
false	*ATF does't contain this variable*

2.5 Dtp_Get_State_Info

Returns the content of a state file given by SZ Sachzeichnungsnummer, MLFB moduletype and ProcessStep. The info is return in the sInfo string and is a pipe delimited string with the info version_1|readytocheck_1|...version_N|readytocheck_N.

readytocheck = 0 RELEASED

readytocheck = 1 READYTOCHECK

Declaration

public **bool** **Dtp_Get_State_Info**(**string** *sSachzeichnungsnummer*, **string** *sMlfb*,
string *sProcessStep*, out **string[]** *sVersions*,
out **string[]** *sVersionInfos*, out **string** *sError*)

Summary:

Dtp_Get_State_Info

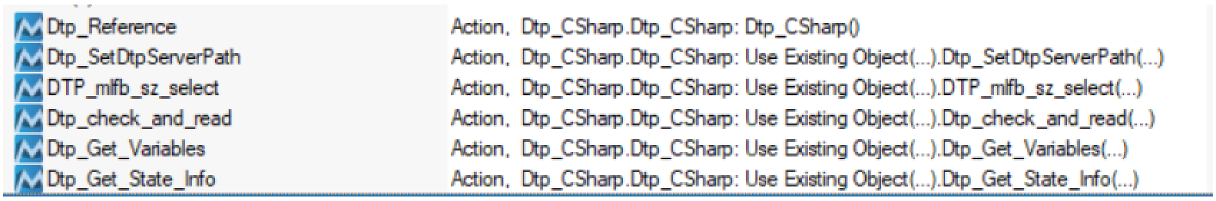
PARAMETERS	DESCRIPTION
sSachzeichnungsnummer	*This parameter is referenced to a valid "Sachzeichnungsnummer" in the dtp path.*
sMlfb	*This parameter is referenced to a valid "Mlfb" in the dtp path.*
sProcessStep	*This parameter is referenced to a valid "ProcessStep" depends on the state of ATF file.* *sVersions: Output: the array is filled by the action and contains the found versions*
sVersions	*Output: the array is filled by the action and contains the found versions*
sVersionInfos	*Output: the array is filled by the action and contains the found versions and also a timestamp of the last changes of the version, also the current dtp state is provided. The timestamp is YYMMDDhhmmss*
sError	*Error information*
RETURN RESULT	**DESCRIPTION**
true	*OK*
false	*No valid state file found*

3 Code Example in C#

```
string sDtpServerPath = "L:\\dtp_wall\\dtp\\DTP_DATA";
int iReady_to_check = 1;
string sSachzeichnungsnummer = "A2C194994";
string sMlfb = "A2C1949941100";
string sProcessStep = "EOL";
string sVersion = "";
int iForce_to_read = 1;
int iCopyFlag = 0;
string sLocalDriveName = "D";
string sData_Path;
int iN_Variables;
string sSelectedSachzeichnungsnummer;
string sSelectedMlfb;
string sSelectedVersion;
string[] sVersions;
string[] sVersionInfos;
string sVariableName = "PROUDCT_PART_NUMBER_FIN_A2C";
string sVariableText;
string[] sarrVariableName = new string[2] { "PROUDCT_PART_NUMBER_FIN_A2C", "HW_VERSION" };
string[] sarrVariableText = new string[2];
List<string> lstVariableName = new List<string> { "PROUDCT_PART_NUMBER_FIN_A2C",
"HW_VERSION" };
List<string> lstVariableText = new List<string>();
string sError;

Dtp_CSharp.Dtp_CSharp dtp = new Dtp_CSharp.Dtp_CSharp();
dtp.Dtp_SetDtpServerPath(sDtpServerPath, out sError);
dtp.DTP_mlfb_sz_select(iReady_to_check, sSachzeichnungsnummer, sProcessStep, out
sSelectedSachzeichnungsnummer, out sSelectedMlfb, out sSelectedVersion, out sError);

dtp.Dtp_check_and_read(iReady_to_check, sSachzeichnungsnummer, sMlfb, sProcessStep,
sVersion, iForce_to_read, iCopyFlag, sLocalDriveName, out sData_Path, out iN_Variables, out
sError);

dtp.Dtp_Get_Variable(sVariableName, out sVariableText, out sError);
dtp.Dtp_Get_Variables(sarrVariableName, out sarrVariableText, out sError);
dtp.Dtp_Get_Variables(lstVariableName, out lstVariableText, out sError);
dtp.Dtp_Get_State_Info(sSachzeichnungsnummer, sMlfb, sProcessStep, out sVersions, out
sVersionInfos, out sError);
```

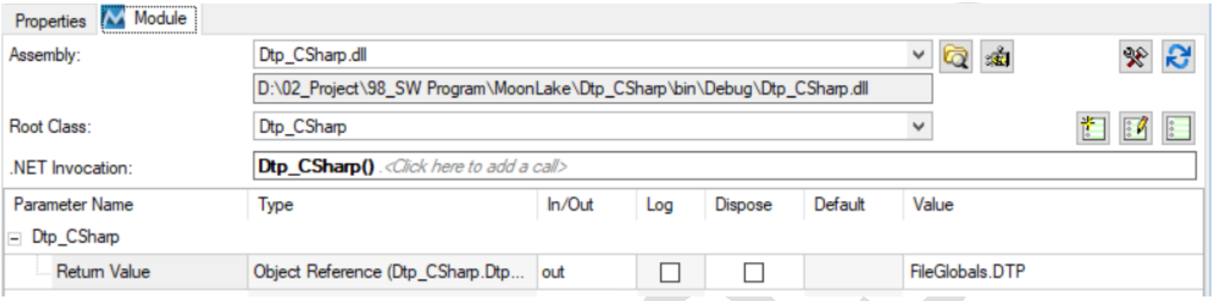
4 Code Example in TestStand

Main



Dtp_Reference	Action, Dtp_CSharp.Dtp_CSharp: Dtp_CSharp()
Dtp_SetDtpServerPath	Action, Dtp_CSharp.Dtp_CSharp: Use Existing Object(...).Dtp_SetDtpServerPath(...)
DTP_mlfb_sz_select	Action, Dtp_CSharp.Dtp_CSharp: Use Existing Object(...).DTP_mlfb_sz_select(...)
Dtp_check_and_read	Action, Dtp_CSharp.Dtp_CSharp: Use Existing Object(...).Dtp_check_and_read(...)
Dtp_Get_Variables	Action, Dtp_CSharp.Dtp_CSharp: Use Existing Object(...).Dtp_Get_Variables(...)
Dtp_Get_State_Info	Action, Dtp_CSharp.Dtp_CSharp: Use Existing Object(...).Dtp_Get_State_Info(...)

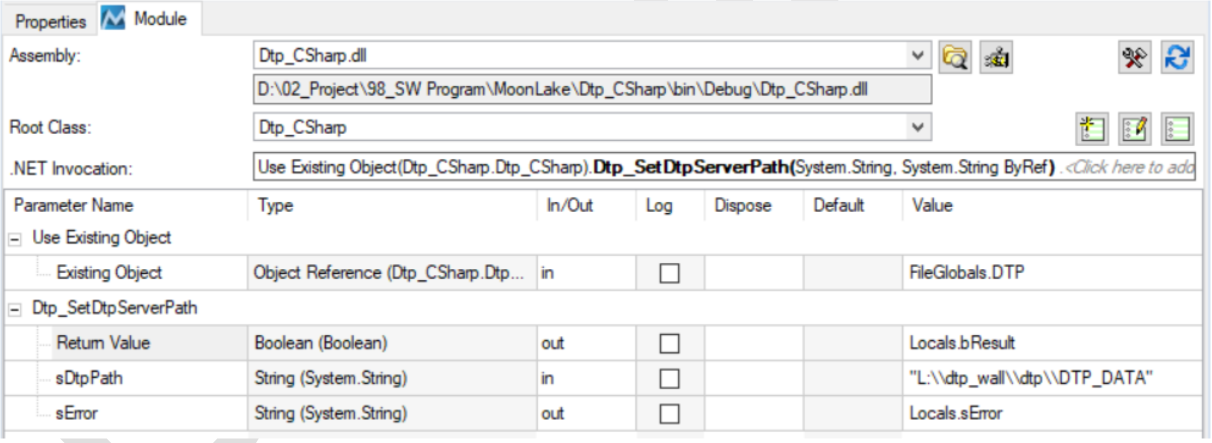
Dtp_Reference



Assembly:	Dtp_CSharp.dll D:\02_Project\98_SW Program\MoonLake\Dtp_CSharp\bin\Debug\Dtp_CSharp.dll
Root Class:	Dtp_CSharp
.NET Invocation:	**Dtp_CSharp()** . <Click here to add a call>

Parameter Name	Type	In/Out	Log	Dispose	Default	Value
⊟ Dtp_CSharp						
Return Value	Object Reference (Dtp_CSharp.Dtp...	out	□	□		FileGlobals.DTP

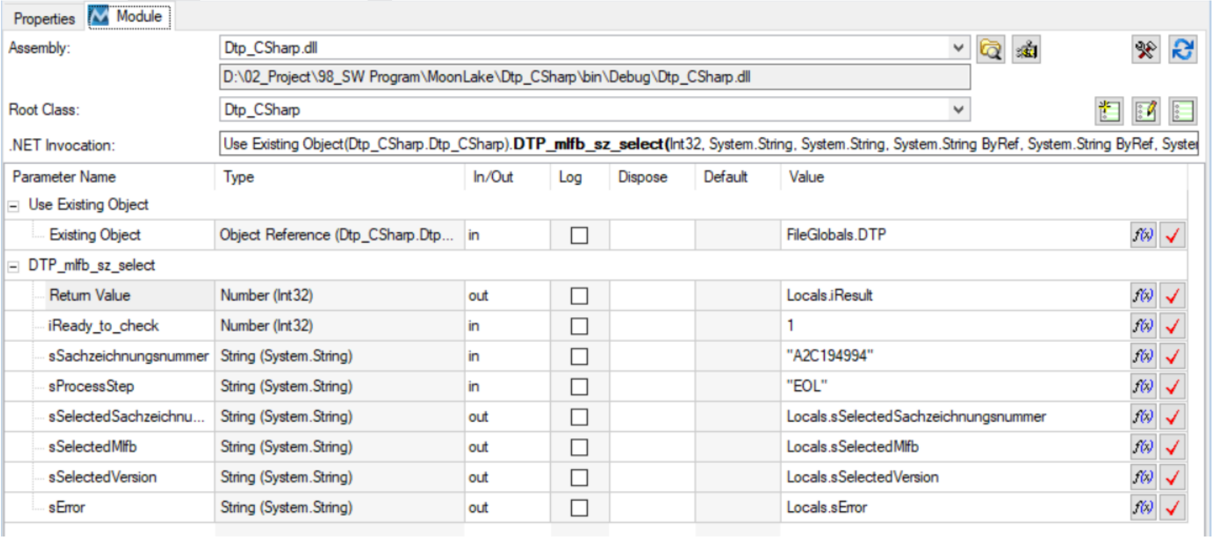
Dtp_SetDtpServerPath



Assembly:	Dtp_CSharp.dll D:\02_Project\98_SW Program\MoonLake\Dtp_CSharp\bin\Debug\Dtp_CSharp.dll
Root Class:	Dtp_CSharp
.NET Invocation:	Use Existing Object(Dtp_CSharp.Dtp_CSharp).**Dtp_SetDtpServerPath(**System.String, System.String ByRef**)** . <Click here to add

Parameter Name	Type	In/Out	Log	Dispose	Default	Value
⊟ Use Existing Object						
Existing Object	Object Reference (Dtp_CSharp.Dtp...	in	□			FileGlobals.DTP
⊟ Dtp_SetDtpServerPath						
Return Value	Boolean (Boolean)	out	□			Locals.bResult
sDtpPath	String (System.String)	in	□			"L:\\dtp_wall\\dtp\\DTP_DATA"
sError	String (System.String)	out	□			Locals.sError

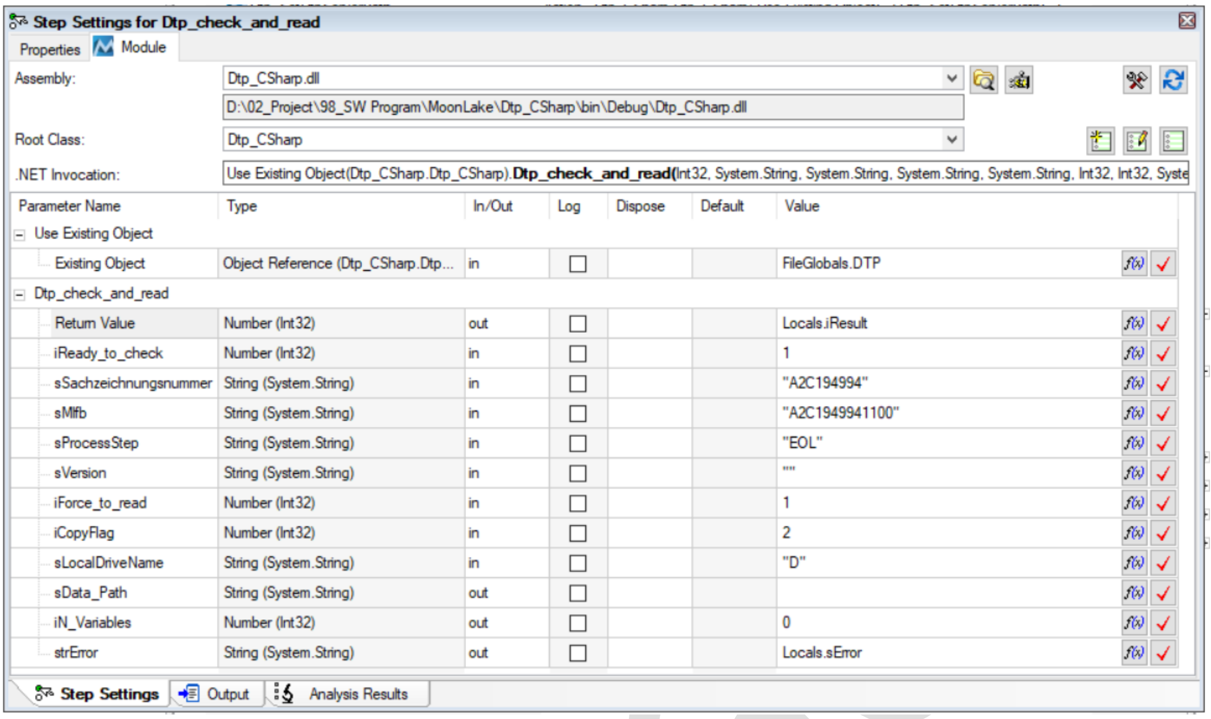
DTP_mlfb_sz_select



Assembly:	Dtp_CSharp.dll D:\02_Project\98_SW Program\MoonLake\Dtp_CSharp\bin\Debug\Dtp_CSharp.dll
Root Class:	Dtp_CSharp
.NET Invocation:	Use Existing Object(Dtp_CSharp.Dtp_CSharp).**DTP_mlfb_sz_select(**Int32, System.String, System.String, System.String ByRef, System.String ByRef, Syste

Parameter Name	Type	In/Out	Log	Dispose	Default	Value
⊟ Use Existing Object						
Existing Object	Object Reference (Dtp_CSharp.Dtp...	in	□			FileGlobals.DTP
⊟ DTP_mlfb_sz_select						
Return Value	Number (Int32)	out	□			Locals.iResult
iReady_to_check	Number (Int32)	in	□			1
sSachzeichnungsnummer	String (System.String)	in	□			"A2C194994"
sProcessStep	String (System.String)	in	□			"EOL"
sSelectedSachzeichnu...	String (System.String)	out	□			Locals.sSelectedSachzeichnungsnummer
sSelectedMlfb	String (System.String)	out	□			Locals.sSelectedMlfb
sSelectedVersion	String (System.String)	out	□			Locals.sSelectedVersion
sError	String (System.String)	out	□			Locals.sError

Dtp_check_and_read



Step Settings for Dtp_check_and_read

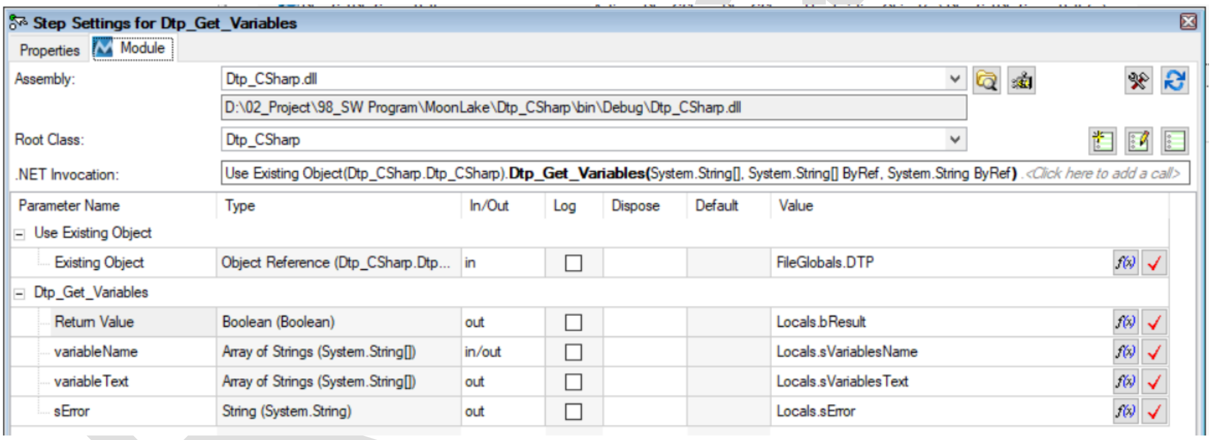
Assembly: Dtp_CSharp.dll
D:\02_Project\98_SW Program\MoonLake\Dtp_CSharp\bin\Debug\Dtp_CSharp.dll

Root Class: Dtp_CSharp

.NET Invocation: Use Existing Object(Dtp_CSharp.Dtp_CSharp).**Dtp_check_and_read(**Int32, System.String, System.String, System.String, System.String, Int32, Int32, Syste

Parameter Name	Type	In/Out	Log	Dispose	Default	Value
Use Existing Object						
Existing Object	Object Reference (Dtp_CSharp.Dtp...	in	□			FileGlobals.DTP
Dtp_check_and_read						
Return Value	Number (Int32)	out	□			Locals.iResult
iReady_to_check	Number (Int32)	in	□			1
sSachzeichnungsnummer	String (System.String)	in	□			"A2C194994"
sMifb	String (System.String)	in	□			"A2C1949941100"
sProcessStep	String (System.String)	in	□			"EOL"
sVersion	String (System.String)	in	□			""
iForce_to_read	Number (Int32)	in	□			1
iCopyFlag	Number (Int32)	in	□			2
sLocalDriveName	String (System.String)	in	□			"D"
sData_Path	String (System.String)	out	□			
iN_Variables	Number (Int32)	out	□			0
strError	String (System.String)	out	□			Locals.sError

Dtp_Get_Variables



Step Settings for Dtp_Get_Variables

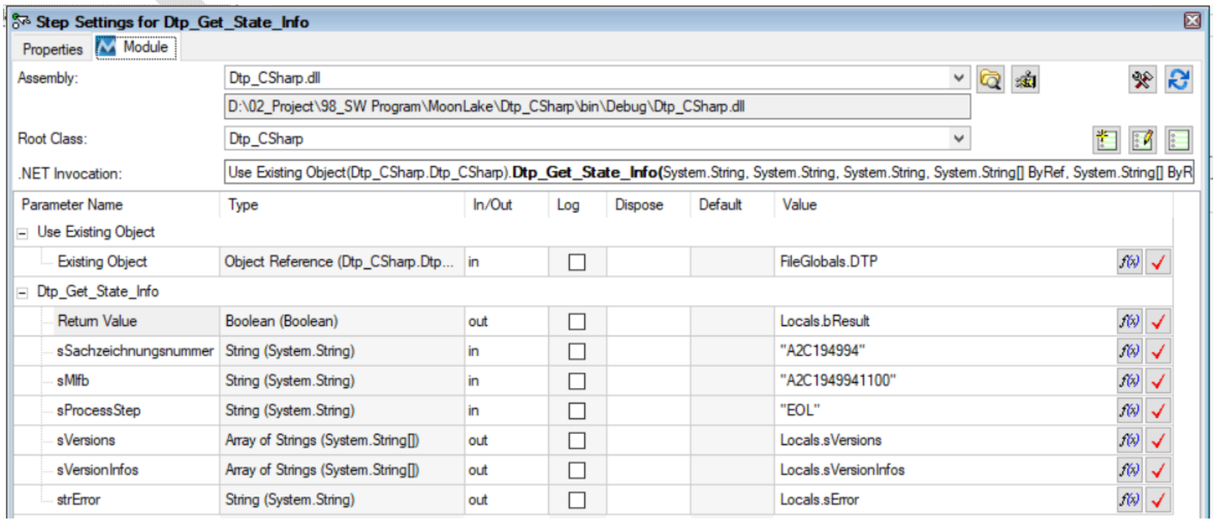
Assembly: Dtp_CSharp.dll
D:\02_Project\98_SW Program\MoonLake\Dtp_CSharp\bin\Debug\Dtp_CSharp.dll

Root Class: Dtp_CSharp

.NET Invocation: Use Existing Object(Dtp_CSharp.Dtp_CSharp).**Dtp_Get_Variables(**System.String[], System.String[] ByRef, System.String ByRef**)** <Click here to add a call>

Parameter Name	Type	In/Out	Log	Dispose	Default	Value
Use Existing Object						
Existing Object	Object Reference (Dtp_CSharp.Dtp...	in	□			FileGlobals.DTP
Dtp_Get_Variables						
Return Value	Boolean (Boolean)	out	□			Locals.bResult
variableName	Array of Strings (System.String[])	in/out	□			Locals.sVariablesName
variableText	Array of Strings (System.String[])	out	□			Locals.sVariablesText
sError	String (System.String)	out	□			Locals.sError

Dtp_Get_State_Info



Step Settings for Dtp_Get_State_Info

Assembly: Dtp_CSharp.dll
D:\02_Project\98_SW Program\MoonLake\Dtp_CSharp\bin\Debug\Dtp_CSharp.dll

Root Class: Dtp_CSharp

.NET Invocation: Use Existing Object(Dtp_CSharp.Dtp_CSharp).**Dtp_Get_State_Info(**System.String, System.String, System.String, System.String[] ByRef, System.String[] ByR

Parameter Name	Type	In/Out	Log	Dispose	Default	Value
Use Existing Object						
Existing Object	Object Reference (Dtp_CSharp.Dtp...	in	□			FileGlobals.DTP
Dtp_Get_State_Info						
Return Value	Boolean (Boolean)	out	□			Locals.bResult
sSachzeichnungsnummer	String (System.String)	in	□			"A2C194994"
sMifb	String (System.String)	in	□			"A2C1949941100"
sProcessStep	String (System.String)	in	□			"EOL"
sVersions	Array of Strings (System.String[])	out	□			Locals.sVersions
sVersionInfos	Array of Strings (System.String[])	out	□			Locals.sVersionInfos
strError	String (System.String)	out	□			Locals.sError