

---

# Appendix

## Actors

### 1. Display Actors

Actor Name	Available as a scripting extension	Comments
ALARMANN	Yes	
ALARMSUM	Yes	
ATTRIBS	Yes	
BACK_OV	No	
BOX_PS	Yes	
BOX_STAT	Yes	
CUSTSAVE	Yes	
CHGZONE	No	
DETAIL	Yes	
DET_PAGE	Yes	
EHR	Yes	
INITFLOP	Yes	
GROUP	Yes	
GRP_EDIT	Yes	
MODGROUP	Yes	
GRPTREND	Yes	
HWPTSUM	Yes	
HWY_STAT	Yes	
LM_DIAG	Yes	
MSGSUM	Yes	
MULT_OV	No	
NIM_DIAG	Yes	
NODE_PS	Yes	
OSUMENU	Yes	

---

## Appendix

Actor Name	Available as a scripting extension	Comments
OVERLAY	No	
OVERVIEW	Yes	
OVW_EDIT	Yes	
PMM_DIAG	Yes	
PM_STAT	Yes	
PROCMOD	Yes	
PVR	Yes	
RTJ	Yes	
REP_MENU	Yes	
SCHEM	Yes	
TITLSUM	Yes	
TRENDOVW	Yes	
IOM_DIAG	Yes	
UCN_STAT	Yes	
UNIT_PS	Yes	
UNITSUM	Yes	
UNITTRND	Yes	
USAGE_PS	Yes	
CANCLPRT	Yes	
CLEAR_CZ	No	
PROMPT_C	Yes	Outputs the given error message to the status bar on the window.
CLR_SCRN	No	
CROSSCRN	No	
DMD_UPD	No	
DSP_FILE	Yes	

---

## Appendix

Actor Name	Available as a scripting extension	Comments
ENT_EXEC	Yes	A dialog box is displayed to handle what would have appeared in the message area of the US display.
FRM_SCRN	No	
KEY	No	
MOVE	Yes	
REM_STR	No	
PROMPT	Yes	Outputs the given error message to the status bar on the window.
PALETTE	No	
PRT_FILE	Yes	
QUE_KEY	Yes	
REM_MOVE	No	
PM_CMD	Yes	
UPDATE	No	
USER_CZ	No	

---

## Appendix

### 2. Keyboard Event Action

Actor Name	Available as a scripting extension	Comments
ENTER	No	Use 'QUE_KEY(enum("\$button:enter")).

---

## Appendix

### 3. Operator Input Actor

Actor Name	Available as a scripting extension	Comments
R_BOOL	Yes	
R_DATE	Yes	
R_DUR	Yes	
R_TIME	Yes	
R_ENT	Yes	
R_ENT_N	Yes	
R_ENM	No	
R_INT	Yes	
R_PAR	Yes	
R_REAL	Yes	
R_SENM	Yes	
R_SENM_N	Yes	
R_STR	Yes	
R_VAR	Yes	<p>The return value of this function in BasicScript is different from the TDC 3000 actor language. In TDC 3000, it is the internal variable identifier. In BasicScript, the return value is the external variable identifier. Use the GetVar function to obtain a var object after reading the variable.</p> <p>For example:</p> <p>S_VAR(VAR01, R_VAR VAR(VAR(1, 2, 5, "enter", TRUE, 0))) from a TDC 3000 picture becomes SET DISPDB.VAR01 = GETVAR(R_VAR(1, 2, 5, "enter", TRUE, 0) in BasicScript.</p>

---

## Appendix

### **4. Store To DDB**

<b>Actor Name</b>	<b>Available as a scripting extension</b>	<b>Comments</b>
S_BOOL	No	Use the BasicScript assignment statement. For example: lcn.a100.pv = TRUE
S_DATE	No	Use the BasicScript assignment statement. For example: S_DATE(DATIME1, C_DATTIM(12-31-96,11:30:32)) from a TDC 3000 picture becomes DISPDB.DATIME1 = (#12/31/1996# + #11:30:32#).
S_DUR	No	Use the BasicScript assignment statement.
S_TIME	No	Use the BasicScript assignment statement
S_ENT	No	Use the BasicScript assignment statement. For example: S_ENT(ENT01, G_ENT(ENT02)) from a TDC 3000 picture becomes DISPDB.ENT01.EXTERNAL = DISPDB.ENT02.EXTERNAL.
S_INT	No	Use the BasicScript assignment statement. For example: S_INT(INT01, G_INT(INT02)) from a TDC 3000 picture becomes DISPDB.INT01 = DISPDB.INT02.
S_REAL	No	Use the BasicScript assignment statement.
S_SENM	No	Use the BasicScript assignment statement.
S_STR	No	Use the BasicScript assignment statement.
S_VAR	No	Use the BasicScript assignment statement.

---

## Appendix

### 5. Read Data And Store into DDB

Actor Name	Available as a scripting extension	Comments
RS_LOC	No	Use the BasicScript assignment statement with the appropriate user input function. For example: RS _LOC(INT01, 1, 2 , 5, "enter", TRUE, 0)) from a TDC 3000 picture becomes DISPDB.INT01 = R_INT(1, 2, 5, "enter", TRUE, 0).

---

## Appendix

### **6. Read From DDB**

<b>Actor Name</b>	<b>Available as a scripting extension</b>	<b>Comments</b>
G_BOOL	No	See discussion of the DispDB built-in object.
G_DATETIME	No	See discussion of the DispDB built-in object.
G_ENT	No	See discussion of the DispDB built-in object.
G_INT	No	See discussion of the DispDB built-in object.
G_REAL	No	See discussion of the DispDB built-in object.
G_SENM	No	See discussion of the DispDB built-in object.
G_STR	No	See discussion of the DispDB built-in object.
G_VAR	No	See discussion of the DispDB built-in object.



---

## Appendix

### **7. Store To DDB**

<b>Actor Name</b>	<b>Available as a scripting extension</b>	<b>Comments</b>
SS_BOOL	No	See discussion of the DispDB built-in object.
SS_DATE	No	See discussion of the DispDB built-in object.
SS_DUR	No	See discussion of the DispDB built-in object.
SS_TIME	No	See discussion of the DispDB built-in object.
SS_ENT	No	See discussion of the DispDB built-in object.
SS_INT	No	See discussion of the DispDB built-in object.
SS_REAL	No	See discussion of the DispDB built-in object.
SS_STR	No	See discussion of the DispDB built-in object.
SS_VAR	No	See discussion of the DispDB built-in object.
EQ_LIST	No	

---

## Appendix

### 8. Read Data And Store into System

Actor Name	Available as a scripting extension	Comments
RS_SYS	No	

---

## Appendix

### 9. Read From DDB

Actor Name	Available as a scripting extension	Comments
GS_BOOL	No	See discussion of the DispDB built-in object.
GS_DATIME	No	See discussion of the DispDB built-in object.
GS_ENT	No	See discussion of the DispDB built-in object.
GS_INT	No	See discussion of the DispDB built-in object.
GS_REAL	No	See discussion of the DispDB built-in object.
GS_SENM	No	See discussion of the DispDB built-in object.
GS_STR	No	See discussion of the DispDB built-in object.
GS_VAR	No	See discussion of the DispDB built-in object.  For example: S_VAR(VAR01, GS_VAR(AMPT.CODSTN(1)) from a TDC 3000 picture becomes Set DISPDB.VAR01 = GetVar(AMPT.CODSTN(1)).
GS_VAR_S	No	See discussion of the DispDB built-in object.

---

## Appendix

### **10. Range Check**

<b>Actor Name</b>	<b>Available as a scripting extension</b>	<b>Comments</b>
RNG_INT	Yes	
RNG_REAL	Yes	
RNG_STR	Yes	
RNG_VAR	No	
RNG_LOC	No	

---

## Appendix

### 11. General Actor

Actor Name	Available as a scripting extension	Comments
C_DATTIM	No	Use the BasicScript '+' operator.
C_V_ENUM	No	The return value of this function in BasicScript is different from the TDC 3000 actor language. In TDC 3000, it is the internal variable identifier. In BasicScript, the return value is the external variable identifier. Example: S_VAR(VAR01, C_VAR(G_ENT(ENT01), 0, .PV, MODE:AUTO) from a TDC 3000 picture becomes SET DISPDB.VAR01 = GETVAR(C_VAR(DISPDB,ENT01.EXTERNAL, 0, ".PV", ENUM("MODE:AUTO"))) in BasicScript.
C_VAR	No	Same comment as C_V_ENUM. Example: S_VAR(VAR01, C_VAR(A100, 0, .PV, -9999) from a TDC 3000 picture becomes SET DISPDB.VAR01 = GETVAR(C_VAR("A100", 0, ".PV", -9999)) in BasicScript.
CONCAT	No	Use the BasicScript '+' operator. For example: CONCAT("aaaa", "bbbb") from a TDC 3000 picture becomes "aaaa" + "bbbb".
DELAY	Yes	
EXTR_ENT	Yes	
EXTR_ID	Yes	
NOP	Yes	
TDC_OPTION	No	

---

## Appendix

### 12. Trend

Actor Name	Available as a scripting extension	Comments
TR_ADD	No	Use the AddTrace method of a Trend control.
TR_CLINE	No	Use the HairlineCursor and HairlineReadout properties of a Trend control. Note that the Hairline functionality of the Trend control replaces the centerline of the TDC 3000 trend phantom.
TR_DATA	No	Use the DataSource property of a Trend control.
TR_DEL	No	Use the DeleteTrace property of a Trend control.
TR_RANGE	No	Use the YRangeHigh and YRangeLow properties of a Trend control.
TR_SCALE	No	Use the YScaleHigh and YScaleLow properties of a Trend control.
TR_SCRLL	No	Use the RuntimeScrolling, ScrollBackTime, ScrollForwardTime and YScaleLow properties of a Trend control.
TR_TIME	No	Use the TimeBase and YScaleLow properties of a Trend control.

---

## Appendix

### 13. Arithmetic

Actor Name	Available as a scripting extension	Comments
ADD_I	No	Use the BasicScript '+' operator.
SUB_I	No	Use the BasicScript '-' operator.
MUL_I	No	Use the BasicScript '*' operator.
DIV_I	No	Use the BasicScript '/' operator.
MOD_I	No	Use the BasicScript 'mod' operator.
NEG_I	No	Use the BasicScript unary '-' operator.
ADD_R	No	Use the BasicScript '+' operator.
SUB_R	No	Use the BasicScript '-' operator..
MUL_R	No	Use the BasicScript '*' operator.
DIV_R	No	Use the BasicScript '/' operator.
NEG_R	No	Use the BasicScript unary '-' operator.

---

## Appendix

### 14. Logical

Actor Name	Available as a scripting extension	Comments
AND	No	Use the BasicScript unary 'AND' operator.
OR	No	Use the BasicScript unary 'OR' operator.
XOR	No	Use the BasicScript unary 'XOR' operator.
NOT	No	Use the BasicScript unary 'NOT' operator.



---

## Appendix

### **15. Comparative**

<b>Actor Name</b>	<b>Available as a scripting extension</b>	<b>Comments</b>
CMP_I	No	Use the BasicScript comparison operator.
CMP_R	No	Use the BasicScript comparison operator.
CMP_S	No	Use the BasicScript comparison operator.
CMP_E	No	Use the BasicScript comparison operator.

---

## Appendix

### 16 Conditional

Actor Name	Available as a scripting extension	Comments
IF	No	Use the BasicScript "if" statement. For example: IF (CMP_R(GS_REAL(A100.PV), GT, 10)); SS_BOOL(A100.PV, TRUE); ELSE; SS_BOOL(A100.PV, FALSE); ENDIF in a TDC 3000 picture becomes if (LCN.A100.PV >= 10)) then LCN.A100.PV = TRUE else LCN.A100.PV = FALSE end if

---

## Appendix

### 17. Conversion

Actor Name	Available as a scripting extension	Comments
FLOAT	No	Use the BasicScript "sng" function.
CNV_I	Yes	
IE_ENT	No	Use the 'external' property of an entity DDB item. For example: S_STR(STRING01, IE_ENT(ENT01)) in a TDC 3000 picture becomes DISPDB.STRING01= DISPDB.ENT01.EXTERNAL.
IE_VAR_E	No	Use the 'external' property of an entity DDB item. For example: S_STR(STRING01, IE_VAR_E(VAR01)) in a TDC 3000 picture becomes DISPDB.STRING01 = EXTR_ENT(DISPDB.VAR01.NAME).
IE_VAR_P	No	Use the EXTR_PAR function on the value of the external property of an VAR DDB item. For example: S_STR(STRING01, IE_VAR_P(VAR01)) in a TDC 3000 picture becomes DISPDB.STRING01 = EXTR_PAR(DISPDB.VAR01.EXTERNAL) (EXTR_PAR is a new function in BasicScript. It does not exist on the TDC 3000).
IE_VAR	No	Use the 'external' property of a VAR DDB item. For example: S_STR(STRING01, IE_VAR(ENT01)) in a TDC 3000 picture becomes DISPDB.STRING01 = DISPDB.VAR01.EXTERNAL.

---

## Appendix

ROUND	Yes	
TRUNC	No	Translated into Basic "FIX".
EI_ENT	No	Use the external property of an ENT DDB item. For example: S_ENT(ENT01, EI_ENT(CONCAT(G_STR(String01), G_STR(String02)))) in a TDC 3000 picture becomes ENT01.EXTERNAL = DISPDB.STRING01 + DISPDB.STRING02.

---

## Appendix

### **18. Keylock**

<b>Actor Name</b>	<b>Available as a scripting extension</b>	<b>Comments</b>
\$KEYRST	Yes	Actor is renamed. \$KEYRST becomes KEY_RST.
\$KEYCHG	Yes	\$KEYCHG(D42ENGR) becomes KEY_CHG(enum("KEYLEVL:ENGR")).
KEY_ENG	No	

---

## Appendix

### Intrinsic functions

Actor Name	Available as a scripting extension	Comments
BIT_TEST	Yes	
EXTERNAL	No	Use the 'external' property of a parameter object. For example: EXTERNAL(A100.PV) in a TDC 3000 picture becomes LCN.A100.PV.EXTERNAL.
INTERNAL	Yes	Use the 'internal' property of a parameter object. For example: INTERNAL(A100.PV) in a TDC 3000 picture becomes LCN.A100.PV.INTERNAL.
STATUS	Yes	Use the 'status' property of a point object. For example. STATUS(A100.PV) in a TDC 3000 picture becomes LCN.A100.PV.STATUS.

---

## Appendix

### New Functions that exist only on GUS to provide TDC 3000 picture functions

Actor Name	Available as a scripting extension	Comments
TDCFORMAT	Yes	<p>TDC VALUE like formatter that supports INTEGER, REAL, STRING, BOOLEAN, DATE/TIME, and UNKNOWN.</p> <p>TDCFORMAT(data, format string)</p> <p>This function has two arguments. The first is the data to be formatted. The type of the first argument depends on the contents of the format string. The second argument is a string containing a TDC 3000 picture editor format string. The return value is a string containing the formatted data.</p> <p>The type of the format string must match the type of data being formatted. The type of an LCN object is not known until runtime, so you must use the "unknown" or "general" type format strings. If you wish to use a more specific format string for LCN objects, use one of the Basic conversion functions such as csng, cstr, date, or cint on the input argument.</p>
GETTREND	No	
EXTR_PAR	Yes	<p>Extract parameter ID(string) EXTR_PAR(string)</p> <p>This function takes one string type argument. Typically, you will use the external property of a VAR DDB item to obtain this string. The return value is a string containing the parameter name and possibly an index if the parameter is an array.</p>

**Last Page**

---

## Appendix



---

## Appendix