

**ASAM GDI COORDINATOR
C++-Interface
Version 2.8**

Document

Release



**rd electronic GmbH
Zweigstelle Dresden
Bernhardstraße 70
01187 Dresden**

Tel. +49 351-6563-56-4

Internet: <http://www.rd-electronic.de>

Status of Document

| | |
|--------------|-------------|
| | |
| Date: | 16.01.12 |
| Last change: | 30.03.21 |
| Author: | RDE |
| Version: | Version 2.8 |
| Doc-ID: | |
| Status: | Release |
| Type | Document |

Revision History

This revision history shows only major modifications between release versions.

| Date | Author | Filename | Comments |
|----------|----------------|------------------------------|---|
| 18.02.03 | Sven Schneider | GDICoord20.DOC | Initial |
| 06.03.03 | Sven Schneider | GDICoord20.DOC | Draft 0.2 |
| 06.04.03 | Sven Schneider | GDICoord20.DOC | Release 1.0 |
| 13.12.05 | Sven Schneider | GDICoord20.DOC | Update |
| 30.06.06 | Sven Schneider | GICO22API.DOC | GICO 22 API |
| 29.01.07 | Sven Schneider | GICO22API.DOC | GICO 22 API Update |
| 24.09.07 | Sven Schneider | GICO22API.DOC | Bugfix |
| 06.10.08 | Sven Schneider | GICO22API.DOC | Improve description of GDICheckVDP preparation Update Errorcodes |
| 24.10.08 | Sven Schneider | GICO22API.DOC | Add API-Functions |
| 19.12.08 | Sven Schneider | GICO22API.DOC | Update Errorcodes Update API-Functions |
| 04.10.10 | Sven Schneider | GICO22API.DOC | Update Parameter of API-Functions |
| 07.09.11 | SBa | GICO22API.DOC | Alignment via GDI_Attach |
| 28.09.11 | SBa | GICO22API.DOC | Correction of validity of errors of GDI Driver |
| 10.10.11 | SBa | GICO22API.DOC Release 2.2 | Correction of description of GDIERROR [4.3] |
| 16.01.12 | SBa | | Refining of comments of Create-Methods in case of errors |
| 17.11.17 | SBa | | Expansion of error codes - COERR_LICENSE |
| 30.03.21 | SBa | | Company information |

Table of contents

| | | |
|----------|--|-----------|
| 1 | <u>Purpose</u> | 12 |
| 2 | <u>General regulations</u> | 12 |
| 2.1 | The ASAM GDI coordinator | 12 |
| 2.2 | Struct member alignment of the ASAM GDI driver | 12 |
| 2.3 | Call convention of the ASAM GDI driver | 12 |
| 3 | <u>Application Interface</u> | 13 |
| 3.1 | Interface GDICoordinator | 13 |
| 3.1.1 | GDICOCreateGDIWorkSpace | 13 |
| 3.1.2 | GDICOGetGDIWorkSpaceIterator | 15 |
| 3.1.3 | GDICODeleteGDIWorkSpace | 16 |
| 3.1.4 | GDICOStartMonitor | 17 |
| 3.1.5 | GDICOStopMonitor | 18 |
| 3.1.6 | GDICOGetVersion | 19 |
| 3.1.7 | GDICOGetDriverVersion | 20 |
| 3.1.8 | GDICOGetPAVersion | 21 |
| 3.1.9 | GDICOGetPAExtVersion | 22 |
| 3.2 | Interface GDIWorkSpace | 23 |
| 3.2.1 | GDICOPrepareInstances | 23 |
| 3.2.2 | GDICOExtendWorkspace | 24 |
| 3.2.3 | GDICOLoadDriver | 25 |
| 3.2.4 | GDICOUNloadDriver | 27 |
| 3.2.5 | GDICOGetDevFuncByName | 28 |
| 3.2.6 | GDICOGetDevFuncIterator | 29 |
| 3.2.7 | GDICOGetDeviceConnectionIterator | 30 |
| 3.2.8 | GDICOGetDCDModuleIterator | 31 |
| 3.2.9 | GDICODriverIdentify | 32 |
| 3.2.10 | GDICODeleteDeviceConnection | 33 |
| 3.2.11 | GDICOGetGDLError | 34 |
| 3.3 | Interface DCDBaseItem | 35 |
| 3.3.1 | GDICOGetIdentifier | 35 |
| 3.4 | Interface DCDModule | 36 |
| 3.4.1 | GDICOGetDCDDevFuncIterator | 36 |
| 3.4.2 | GDICOGetConnectingData | 37 |
| 3.4.3 | GDICOCreateDeviceConnection | 38 |
| 3.5 | Interface DCDDeviceFunction | 39 |
| 3.5.1 | GDICOGetDCDComObjIterator | 39 |
| 3.5.2 | GDICOGetDCDOperationIterator | 40 |
| 3.6 | Interface DCDComObject | 41 |
| 3.6.1 | GDICOGetProperties | 41 |
| 3.7 | Interface DCDOperation | 42 |
| 3.8 | Interface GDIDeviceConnection | 43 |

| | | |
|-------------|------------------------------------|-----------|
| 3.8.1 | GDICOGetInstanceName | 43 |
| 3.8.2 | GDICOVDIdentify | 44 |
| 3.8.3 | GDICOVDStatus | 45 |
| 3.8.4 | GDICOCheckVDPPreparation | 46 |
| 3.8.5 | GDICOSTopWorking | 47 |
| 3.8.6 | GDICOCreateDeviceFunction | 48 |
| 3.8.7 | GDICOGetCreateParameterData | 50 |
| 3.8.8 | GDICODeleteDeviceFunction | 51 |
| 3.9 | Interface GDIDeviceFunction | 52 |
| 3.9.1 | GDICOGetInstanceName | 52 |
| 3.9.2 | GDICOCreateComObject | 53 |
| 3.9.3 | GDICOCreateOperation | 54 |
| 3.9.4 | GDICOGetComObjectByName | 55 |
| 3.9.5 | GDICOGetComObjectIterator | 56 |
| 3.9.6 | GDICOGetOperationByName | 57 |
| 3.9.7 | GDICOGetOperationIterator | 58 |
| 3.10 | Interface GDIComObject | 59 |
| 3.10.1 | GDICOGetInstanceName | 59 |
| 3.10.2 | GDICOGetInputParameterData | 60 |
| 3.10.3 | GDICOGetOutputParameterData | 61 |
| 3.10.4 | GDICOWrite | 62 |
| 3.10.5 | GDICORead | 63 |
| 3.10.6 | GDICOAcceptOccured | 64 |
| 3.10.7 | GDICOInfReportOccured | 65 |
| 3.11 | Interface GDIOperation | 66 |
| 3.11.1 | GDICOGetInstanceName | 66 |
| 3.11.2 | GDICOGetInputParameterData | 67 |
| 3.11.3 | GDICOGetOutputParameterData | 68 |
| 3.11.4 | GDICOExecute | 69 |
| 3.12 | Call back functions | 70 |
| 3.12.1 | GDIMSGTransferCompleteCO | 70 |
| 3.12.2 | GDIMSGTransferCompleteOp | 71 |
| 3.12.3 | GDIMSGDriverReport | 72 |
| 3.12.4 | GDIMSGException | 73 |
| 3.12.5 | GDIMSGAccept | 74 |
| 3.12.6 | GDIMSGCreateDeviceConnection | 75 |
| 3.12.7 | GDIMSGCreateDeviceFunction | 76 |
| 3.12.8 | GDIMSGCreateComObject | 77 |
| 3.13 | Interface GDIDataBaseType | 78 |
| 3.13.1 | GDIDATAGetType | 78 |
| 3.13.2 | GDIDATAGetInstanceVersion | 79 |
| 3.13.3 | GDIDATAGetOrdValue | 80 |
| 3.13.4 | GDIDATASetValue | 81 |
| 3.13.5 | GDIDATAGetValue | 82 |
| 3.14 | Interface GDIDataShort | 84 |
| 3.14.1 | GDIDATAGetMaxValue | 84 |
| 3.14.2 | GDIDATAGetMinValue | 85 |
| 3.14.3 | GDIDATAGetStepValue | 86 |
| 3.14.4 | GDIDATASetShortValue | 87 |
| 3.14.5 | GDIDATAGetShortValue | 88 |
| 3.15 | Interface GDIDataUShort | 90 |

| | | |
|-------------|--|------------|
| 3.15.1 | GDIDATAGetMaxValue | 90 |
| 3.15.2 | GDIDATAGetMinValue | 91 |
| 3.15.3 | GDIDATAGetStepValue | 92 |
| 3.15.4 | GDIDATASetUShortValue | 93 |
| 3.15.5 | GDIDATAGetUShortValue | 94 |
| 3.16 | Interface GDIDataULong | 95 |
| 3.16.1 | GDIDATAGetMaxValue | 95 |
| 3.16.2 | GDIDATAGetMinValue | 96 |
| 3.16.3 | GDIDATAGetStepValue | 97 |
| 3.16.4 | GDIDATASetULongValue | 98 |
| 3.16.5 | GDIDATAGetULongValue | 99 |
| 3.17 | Interface GDIDataLong | 100 |
| 3.17.1 | GDIDATAGetMaxValue | 100 |
| 3.17.2 | GDIDATAGetMinValue | 101 |
| 3.17.3 | GDIDATAGetStepValue | 102 |
| 3.17.4 | GDIDATASetLongValue | 103 |
| 3.17.5 | GDIDATAGetLongValue | 104 |
| 3.18 | Interface GDIDataFloat | 106 |
| 3.18.1 | GDIDATAGetMaxValue | 106 |
| 3.18.2 | GDIDATAGetMinValue | 107 |
| 3.18.3 | GDIDATAGetStepValue | 108 |
| 3.18.4 | GDIDATASetFloatValue | 109 |
| 3.18.5 | GDIDATAGetFloatValue | 110 |
| 3.19 | Interface GDIDataDouble | 112 |
| 3.19.1 | GDIDATAGetMaxValue | 112 |
| 3.19.2 | GDIDATAGetMinValue | 113 |
| 3.19.3 | GDIDATAGetStepValue | 114 |
| 3.19.4 | GDIDATASetDoubleValue | 115 |
| 3.19.5 | GDIDATAGetDoubleValue | 116 |
| 3.20 | Interface GDIDataChar | 118 |
| 3.20.1 | GDIDATAGetMaxValue | 118 |
| 3.20.2 | GDIDATAGetMinValue | 119 |
| 3.20.3 | GDIDATAGetStepValue | 120 |
| 3.20.4 | GDIDATASetCharValue | 121 |
| 3.20.5 | GDIDATAGetCharValue | 122 |
| 3.21 | Interface GDIDataUChar | 123 |
| 3.21.1 | GDIDATAGetMaxValue | 123 |
| 3.21.2 | GDIDATAGetMinValue | 124 |
| 3.21.3 | GDIDATAGetStepValue | 125 |
| 3.21.4 | GDIDATASetCharValue | 126 |
| 3.21.5 | GDIDATAGetCharValue | 127 |
| 3.22 | Interface GDIDataBoolean | 128 |
| 3.22.1 | GDIDATASetBooleanValue | 128 |
| 3.22.2 | GDIDATAGetBooleanValue | 129 |
| 3.23 | Interface GDIDataDevFuncReference | 130 |
| 3.23.1 | GDIDATASetDevFunc | 130 |
| 3.23.2 | GDIDATAGetDevFunc | 131 |
| 3.24 | Interface GDIDataEnum | 132 |
| 3.24.1 | GDIDATAGetFirstElement | 132 |
| 3.24.2 | GDIDATAGetNextElement | 133 |

| | | |
|-------------|---|------------|
| 3.25 | Interface GDIDataStructure | 134 |
| 3.25.1 | GDIDATAGetDataTypeliterator | 134 |
| 3.25.2 | GDIDATAGetSubvalueByName | 135 |
| 3.25.3 | GDIDATAGetSubvalueByOrdValue | 136 |
| 3.26 | Interface GDIDataUnion | 138 |
| 3.26.1 | GDIDATAGetDataTypeliterator | 138 |
| 3.26.2 | GDIDataGetSwitchType | 140 |
| 3.26.3 | GDIDATAGetSubvalueByName | 142 |
| 3.26.4 | GDIDATAGetSubvalueByOrdValue | 143 |
| 3.27 | Interface GDIDataSequence | 144 |
| 3.27.1 | GDIDATAGetSubType | 144 |
| 3.27.2 | GDIDATAGetCopySubTypeByIndex | 144 |
| 3.27.3 | GDIDATAGetSubTypeByIndex | 145 |
| 3.27.4 | GDIDATAGetSequenceType | 146 |
| 3.27.5 | GDIDATAGetMaxElementNumber | 148 |
| 3.27.6 | GDIDATAGetCurrentSize | 149 |
| 3.27.7 | GDIDATASetCurrentSize | 150 |
| 3.27.8 | GDIDATAGetElementType | 151 |
| 3.28 | Interface GDIDataArray | 152 |
| 3.28.1 | GDIDATAGetSubType | 152 |
| 3.28.2 | GDIDATAGetCopySubTypeByIndex | 152 |
| 3.28.3 | GDIDATAGetSubTypeByIndex | 152 |
| 3.28.4 | GDIDATAGetMaxElementNumber | 154 |
| 3.28.5 | GDIDATAGetElementType | 155 |
| 3.29 | Interface GDIIterator | 156 |
| 3.29.1 | GDICOGetFirstItem | 156 |
| 3.29.2 | GDICOGetNextItem | 157 |
| 3.30 | Interface WorkSpaceliterator | 158 |
| 3.30.1 | GDICOGetFirstWorkSpace | 158 |
| 3.30.2 | GDICOGetNextWorkSpace | 159 |
| 3.31 | Interface GDIDeviceConnectionliterator | 160 |
| 3.32 | Interface DCDModuleliterator | 161 |
| 3.33 | Interface GDIDevFuncIterator | 162 |
| 3.34 | Interface DCDDevFuncIterator | 163 |
| 3.35 | Interface DCDComObjectIterator | 164 |
| 3.36 | Interface GDIComObjectIterator | 165 |
| 3.37 | Interface DCDOperationIterator | 166 |
| 3.38 | Interface GDIOperationIterator | 167 |
| 3.39 | Interface GDIDataTypeliterator | 168 |
| 4 | Definitions of Datatype | 169 |
| 4.1 | COMOBJECTTYPE | 169 |
| 4.2 | GDIACCRIGHTS | 170 |
| 4.3 | GDIERROR | 171 |
| 4.4 | GDIDITERRORPARAMETER | 172 |

| | | |
|----------|----------------------|------------|
| 4.5 | ACCEPTOCCURED | 173 |
| 4.6 | INFREPORTOCCURED | 174 |
| 4.7 | GDIDATATYPE | 175 |
| 4.8 | DATAINSTANCEVERSION | 176 |
| 4.9 | GDISEQUENCETYPE | 177 |
| 4.10 | GDIENUMERATIONMEMBER | 178 |
| 4.11 | GDICOMMTYPE | 179 |
| 4.12 | GDIIDENT | 180 |
| 4.13 | GDISTAT | 181 |
| 4.14 | INSTANCEITEM | 182 |
| 4.15 | IDENTITEM | 183 |
| 5 | Error codes | 184 |

1 PURPOSE

The ASAM GDI coordinator and the ASAM GDI CORBA server are products of the company **rd electronic gmbh**. This document describes the interface, the characteristics and the behavior of the ASAM GDI coordinator.

2 GENERAL REGULATIONS

2.1 THE ASAM GDI COORDINATOR

Function of the GDI coordinator is to make a connection between GDI driver and application. For this the DCD file is read and interpreted by the coordinator. On base of this DCD file the necessary objects and data areas can be created and administered. The coordinator takes over completely the administration of the GDI instances and executes depending on request independently the transition to the individual GDI operating phases. All data areas of the input and output parameters are held within the address range of the coordinator and also are administered by this.

2.2 STRUCT MEMBER ALIGNMENT OF THE ASAM GDI DRIVER

One byte is defined as the struct member alignment of the GDI driver.

2.3 CALL CONVENTION OF THE ASAM GDI DRIVER

The GDI coordinator / server program uses the cdecl call convention. This is the default calling convention for C and C++ programs. The stack is cleaned up by the caller.

| Element | Implementation |
|----------------------------------|--|
| Argument-passing order | Right to left |
| Stack-maintenance responsibility | Calling function pops the arguments from the stack |
| Name-decoration convention | Underscore character (_) is prefixed to names |
| Case-translation convention | No case translation performed |

3 APPLICATION INTERFACE

3.1 INTERFACE GDICoORDINATOR

3.1.1 GDICOCREATEGDIWORKSPACE

Prototype:

```
short GDICOCreateGDIWorkspace (    GDIWorkspace*& pGDIWorkspaceObj,  
                                   GDICOCBTC pTransferCompleteCO,  
                                   GDICOCBTO pTransferCompleteOp,  
                                   GDICOCBDR pDriverReport,  
                                   GDICOCBEX pExeption,  
                                   GDICOCBAC pAccept,  
                                   GDICOCBCC pCreateDeviceConnection,  
                                   GDICOCBCF pCreateDeviceFunction,  
                                   GDICOCBCO pCreateComObject,  
                                   unsigned long Timeout)
```

Task:

The function creates a GDI Work Space. Within this Work Space the GDI driver is controlled by the coordinator. This Work Space is the work area of all GDI driver functionalities. In each case a GDI driver per Workspace can be used.

Input Parameters:

| | |
|-----------------------------------|--|
| GDICOCBTC pTransferCompleteCO | pointer on GDIMSGTransferCompleteCO |
| GDICOCBTO pTransferCompleteOp | pointer on GDIMSGTransferCompleteOp |
| GDICOCBDR pDriverReport | pointer on GDIMSGDriverReport |
| GDICOCBEX pExeption | pointer on GDIMSGException |
| GDICOCBAC pAccept | pointer on GDIMSGAccept |
| GDICOCBCC pCreateDeviceConnection | pointer on GDIMSGCreateDeviceConnection |
| GDICOCBCF pCreateDeviceFunction | pointer on GDIMSGCreateDeviceFunction |
| GDICOCBCO pCreateComObject | pointer on GDIMSGCreateComObject |
| unsigned long Timeout | time in milliseconds, those the GDI-API for their calls needs;if the GDI driver function does not return in this Peroid, an error is released by coordinator |

Output Parameters:

GDIWorkSpace*& pGDIWorkSpaceObj reference of the GDI Work Space Object – the reference is only valid if the call was executed faultless

Return value:

Error message of the GDI coordinator

3.1.2 GDICOGETGDIWORKSPACEITERATOR

Prototype:

short GDICOGetGDIWorkSpaceliterator (GDIWorkSpaceliterator*&
pGDIWorkSpaceliterator)

Task:

The function gets a reference of the GDI Work Space Iterator.

Input Parameters:**Output Parameters:**

GDIWorkSpaceliterator*& pGDIWorkSpaceliterator reference of the GDI Work Space
Iterator object

Return value:

Error message of the GDI coordinator

3.1.3 GDICODELETEGDIWORKSPACE

Prototype:

short GDICODEleteGDIWorkSpace (GDIWorkSpace* pGDIWorkSpaceObj)

Task:

The function deletes the GDI Work Space.

The possibly still existing device VD is closed with GDI_Abort. The connection to the control VD is separated with GDI_Conclude (in the event of an error with GDI_Abort). Subsequently, all data areas of the project not eliminated yet are deleted and the driver is unloaded¹.

Input Parameters:

| | |
|--------------------------------|--|
| GDIWorkSpace* pGDIWorkSpaceObj | reference of the GDI Work Space Object |
|--------------------------------|--|

Output Parameters:**Return value:**

Error message of the GDI coordinator

¹ The same GDI driver with all its instances of a other Work Spaces is not affected.

3.1.4 GDICOSTARTMONITOR

Prototype:

```
void GDICOSTartMonitor (const char* Monitorfile)
```

Task:

The function starts the monitoring process.

Input Parameters:

char* Monitorfile Name of the monitor file

Output Parameters:**Return value:**

3.1.5 GDICOSTOPMONITOR

Prototype:

```
void GDICOSTopMonitor ()
```

Task:

The function stops the monitoring process.

Input Parameters:**Output Parameters:****Return value:**

3.1.6 GDICOGETVERSION

Prototype:

```
void GDICOGetVersion(    unsigned short* punMaster,  
                        unsigned short* punMajor,  
                        unsigned short* punMinor,  
                        unsigned short* punBuild)
```

Task:

Gets the version of the GDI coordinator.

Input Parameters:**Output Parameters:**

| | |
|---------------------------|----------------|
| unsigned short* punMaster | Version master |
| unsigned short* punMajor | Version major |
| unsigned short* punMinor | Version minor |
| unsigned short* punBuild | Version build |

Return value:

Error message of the GDI coordinator

3.1.7 GDICOGOETDRIVERVERSION

Prototype:

```
short GDICOGOETDRIVERVERSION(    const char* pszDriverName,  
                                  const char* pszDCDName,  
                                  unsigned short* punMaster,  
                                  unsigned short* punMajor,  
                                  unsigned short* punMinor,  
                                  unsigned short* punBuild)
```

Task:

Gets the version of the GDI driver.

Input Parameters:

| | |
|---------------------------|-------------------------|
| const char* pszDriverName | Name of the GDI driver. |
| const char* pszDCDName | Name of the DCD file. |

Output Parameters:

| | |
|---------------------------|----------------|
| unsigned short* punMaster | Version master |
| unsigned short* punMajor | Version major |
| unsigned short* punMinor | Version minor |
| unsigned short* punBuild | Version build |

Return value:

Error message of the GDI coordinator

3.1.8 GDICOGETPAVERSION

Prototype:

```
short GDICOGetPAVersion( unsigned short* punMaster,  
                          unsigned short* punMajor,  
                          unsigned short* punMinor,  
                          unsigned short* punBuild)
```

Task:

Gets the version of the platformadapter.

Input Parameters:**Output Parameters:**

| | |
|---------------------------|----------------|
| unsigned short* punMaster | Version master |
| unsigned short* punMajor | Version major |
| unsigned short* punMinor | Version minor |
| unsigned short* punBuild | Version build |

Return value:

Error message of the GDI coordinator

3.1.9 GDICOGETPAEXTVERSION

Prototype:

```
short GDICOGetPAExtVersion(    const char* pszExtensionName,  
                               unsigned short* punMaster,  
                               unsigned short* punMajor,  
                               unsigned short* punMinor,  
                               unsigned short* punBuild);
```

Task:

Gets the version of the platformadapter.

Input Parameters:

| | |
|------------------------------|---------------------------------------|
| const char* pszExtensionName | Name of the platformadapter extension |
|------------------------------|---------------------------------------|

Output Parameters:

| | |
|---------------------------|----------------|
| unsigned short* punMaster | Version master |
| unsigned short* punMajor | Version major |
| unsigned short* punMinor | Version minor |
| unsigned short* punBuild | Version build |

Return value:

Error message of the GDI coordinator

3.2 INTERFACE GDIWORKSPACE

3.2.1 GDICOPREPAREINSTANCES

Prototype:

```
short GDICOPrepareInstances ( const char* InstanceDescription,  
                              GDIERROR* pExecutionErr ,  
                              unsigned short StructMemberAlignment)
```

Task:

The function creates all GDI driver Instances, which are defined by the InstanceDescription².

The function opens transferred DCD for it and loads the appropriate driver. Here the DCD is checked for plausibility and syntax. If the DCD file is error-free and could the driver be loaded, the function *GDI_Attach* is called and the necessary points of the GDI call-back functions are transferred. Subsequently, the creation of the control VD takes place.

After this the coordinator creates all driver Instances, which are defined by the InstanceDescription.

In a case of an error the coordinator deletes all created instances and unloads the GDI driver. The possibly still existing device VD is closed with *GDI_Abort*. The coordinator deletes all data areas of the Work Space³.

The function may be implemented successfully in each GDI Work space only after the call of *GDICOCreateGDIWorkSpace* or *GDICOUncloadDriver*.

Input Parameters:

| | |
|--------------------------------------|---|
| char* InstanceDescription | Name of the parameterization data (e.g. file name) |
| unsigned short StructMemberAlignment | Struct member alignment of the GDI driver (default 1) |

Output Parameters:

| | |
|-------------------------|---------------------------------------|
| GDIERROR* pExecutionErr | error message of the GDI-API function |
|-------------------------|---------------------------------------|

Return value:

Error message of the GDI coordinator

² The same GDI driver with all its instances of a other Work Spaces is not affected.

³ The same GDI driver with all its instances of a other Work Spaces is not affected.

3.2.2 GDICOEXTENDWORKSPACE

Prototype:

```
short GDICOExtendWorkspace ( const char* InstanceDescription,  
                             GDIERROR* pExecutionErr,  
                             unsigned short StructMemberAlignment)
```

Task:

The function extends the Wokspace.

It creates new Instances of the GDI driver, which are defined by the InstanceDescription⁴.

In a case of an error the coordinator deletes all new created instances. .

The function may be implemented successfully in each GDI Work space only after the call of *GDICOCreateGDIWorkSpace* and *GDICOPrepareInstances* or *GDICOLoadDriver*.

Input Parameters:

| | |
|--------------------------------------|--|
| char* InstanceDescription | Name of the parameterization data (e.g. file name) |
| unsigned short StructMemberAlignment | Struct member alignment of the GDI driver |

Output Parameters:

| | |
|-------------------------|---------------------------------------|
| GDIERROR* pExecutionErr | error message of the GDI-API function |
|-------------------------|---------------------------------------|

Return value:

Error message of the GDI coordinator

Reference to ASAM GDI Coordinator Services:

gdiCreateSession

⁴ The same GDI driver with all its instances of a other Work Spaces is not affected.

3.2.3 GDICOLoadDRIVER

Prototype:

```
short GDICOLoadDriver (  const char* DCDName,  
                        const char* Drivename,  
                        const char* DITLanguage,  
                        GDIERROR* pExecutionErr,  
                        unsigned short StructMemberAlignment)
```

Task:

The function opens transferred DCD for it and loads the appropriate driver. Here the DCD is checked for plausibility and syntax. If the DCD file is error-free and could the driver be loaded, the function *GDI_Attach* is called and the necessary points of the GDI call-back functions are transferred. The alignment of device driver is overtaken if transmitted. Subsequently, the creation of the control VD and the initialization of the transition functions takes place⁵.

If the country code is set in the parameter *DITLanguage*, the server loads the special DIT file and the standard file *COM_ERR.DIT*.

The function may be implemented successfully in each GDI Work space only after the call of *GDICOCreateGDIWorkSpace* or *GDICOUNloadDriver*.

Input Parameters:

| | |
|--------------------------------------|---|
| char* DCDName | name and path to the project belonging of the DCD file, DCD.H file and DIT files (coordinator uses only path of dcd file) |
| char* Drivename | name and path to the project belonging of the GDI driver |
| char* DITLanguage | contraction which can be used of the DIT designation of the language |
| unsigned short StructMemberAlignment | Struct member alignment of the GDI driver (default 1) |

Output Parameters:

| | |
|-------------------------|---------------------------------------|
| GDIERROR* pExecutionErr | error message of the GDI-API function |
|-------------------------|---------------------------------------|

Return value:

Error message of the GDI coordinator

⁵ The same GDI driver with all its instances of a other Work Spaces is not affected.



3.2.4 GDICOUNLOADDRIVER

Prototype:

short GDICOUNloadDriver ()

Task:

The function unloads the GDI driver and deletes all existing instances⁶. The possibly still existing device VD is closed with GDI_Abort. The connection to the control VD is separated with GDI_Conclude (in the event of an error with GDI_Abort). Subsequently, all data areas of the project not eliminated yet are deleted and the driver is unloaded.

Input Parameters:**Output Parameters:****Return value:**

Error message of the GDI coordinator

⁶ The same GDI driver with all its instances of a other Work Spaces is not affected.

3.2.5 GDICOGETDEVFUNCBYNAME

Prototype:

```
short GDICOGetDevFuncByName (    const char* FuncObjName,  
                                GDIDeviceFunction* pGDIDeviceFunction)
```

Task:

The function gets a reference to an existing function object. The object must have been produced with the function *GDICOCreateDeviceFunction*.

Input Parameters:

| | |
|-------------------|---------------------------|
| char* FuncObjName | Name of expected Instance |
|-------------------|---------------------------|

Output Parameters:

| | |
|---------------------------------------|------------------------------|
| GDIDeviceFunction* pGDIDeviceFunction | reference of the func object |
|---------------------------------------|------------------------------|

Return value:

Error message of the GDI coordinator

3.2.6 GDICOGETDEVFUNCITERATOR

Prototype:

short GDICOGetDevFuncIterator (GDIDevFuncIterator*& pGDIDevFuncIterator)

Task:

The function gets a reference to an iterator, which is used for the output of all set up device functions (function objects) of the Work Space.

Input Parameters:**Output Parameters:**

| | |
|--|---|
| GDIDevFuncIterator*& pGDIDevFuncIterator | reference of the device function iterator |
|--|---|

Return value:

Error message of the GDI coordinator

3.2.7 GDICOGETDEVICECONNECTIONITERATOR

Prototype:

```
short GDICOGetDeviceConnectionIterator ( GDIDeviceConnectionIterator*&  
                                          pGDIDeviceConnectionIterator)
```

Task:

The function gets a reference to an iterator, which is used for the output of all set up VD instances of the Work Space.

Input Parameters:**Output Parameters:**

| | |
|--|--|
| GDIDeviceConnectionIterator*& pGDIDeviceConnectionIterator | reference of the Device connection iterator |
|--|--|

Return value:

Error message of the GDI coordinator

3.2.8 GDICOGETDCDMODULEITERATOR

Prototype:

```
short GDICOGetDCDModuleIterator ( DCDModuleIterator*&  
                                  pGDIModuleIterator)
```

Task:

The function gets a reference to an iterator, which is used for the output of all modules, which are defined by the DCD.

Input Parameters:**Output Parameters:**

DCDModuleIterator*& pGDIModullterator reference of the module iterator

Return value:

Error message of the GDI coordinator

3.2.9 GDICODRIVERIDENTIFY

Prototype:

short GDICODriverIdentify (GDIIDENT* pIdentInformation,
 GDIERROR* pExecutionErr)

Task:

The function calls the GDI driver function *GDI_Identify*.
GDI_Identify delivers data for identification of the device manufacturers .

Input Parameters:**Output Parameters:**

| | |
|-----------------------------|---------------------------------------|
| GDIIDENT* pIdentInformation | Ident Information |
| GDIERROR* pExecutionErr | error message of the GDI-API function |

Return value:

Error message of the GDI coordinator

3.2.10 GDICODELETEDEVICECONNECTION

Prototype:

```
short GDICODEleteDeviceConnection ( GDIDeviceConnection*  
                                     pGDIDeviceConnection,  
                                     GDIERROR* pExecutionErr)
```

Task:

The function deletes the connection to a device, which is supported by the created VD instance. Here the respective VD handle with *GDI_Conclude* will be closed after it with *ClearAllObject* into the status to *Initialized* was brought. All data areas of the VD instance not eliminated yet are deleted.

Input Parameters:

| | |
|---|---|
| GDIDeviceConnection* pGDIDeviceConnection | reference of the object, which represents a created device connection |
|---|---|

Output Parameters:

| | |
|-------------------------|---------------------------------------|
| GDIERROR* pExecutionErr | error message of the GDI-API function |
|-------------------------|---------------------------------------|

Return value:

Error message of the GDI coordinator

3.2.11 GDICOGETGDIERROR

Prototype:

```
short GDICOGetGDIError ( GDIDITERRORPARAMETER& DITErrorParameter,  
                          GDIERROR* pGDIError)
```

Task:

Gets the GDI error codes and the DIT text message. The error codes are set by the application.

Input Parameters:

| | |
|---|---|
| GDIDITERRORPARAMETER& DITErrorParameter | Replace text and error codes of the message |
|---|---|

Output Parameters:

| | |
|---------------------|---|
| GDIERROR* pGDIError | GDI Errorcode and error text by DIT file. |
|---------------------|---|

Return value:

Error message of the GDI coordinator

3.3 INTERFACE DCDBASEITEM

3.3.1 GDICOGETIDENTIFIER

Prototype:

short GDICOGetIdentifier (IDENTITEM* pIdentifier)

Task:

The function returns the name of a DCD Item, which is defined in the dcd file. Items are modules, interfaces, definitions of data types, parameters, attributes or operations.

Input Parameters:

Output Parameters:

IDENTITEM* pIdentifier

Identifier of the item, which is defined in the dcd file

Return value:

Error message of the GDI coordinator

3.4 INTERFACE DCDDMODULE

Basis class

DCDBasItem

Task

The interface contains informations about the modul, which is defined by the selected dcd file.

3.4.1 GDICOGETDCDDDEVFUNCITERATOR

Prototype:

```
short GDICOGetDCDDDevFuncIterator ( DCDDDevFuncIterator*&  
                                     pDevFuncIterator)
```

Task:

The function gets a reference to an iterator, which is used for the output of all device function defined by the dcd file.

Input Parameters:

Output Parameters:

| | |
|---|---|
| DCDDDevFuncIterator*& pGDIDevFuncIterator | reference of the device function iterator |
|---|---|

Return value:

Error message of the GDI coordinator

3.4.2 GDICOGETCONNECTINGDATA

Prototype:

short GDICOGetConnectingData (GDIDataBaseType*& pDataObj)

Task:

The function supplies the reference of the data object of the input parameter of the DCD function *IOFunction*.

Input Parameters:**Output Parameters:**

| | |
|----------------------------|---|
| GDIDataBaseType*& pDataObj | reference of the data object, which describes the input parameter of GDI_Initiate |
|----------------------------|---|

Return value:

Error message of the GDI coordinator

3.4.3 GDICOCREATEDEVICECONNECTION

Prototype:

```
short GDICOCreateDeviceConnection ( GDICOMMTYPE Communication,  
                                     INSTANCEITEM InstanceName,  
                                     GDIDataBaseType* pDataObj,  
                                     GDIDeviceConnection*&  
                                     pGDIDeviceConnection,  
                                     GDIERROR* pExecutionErr)
```

Task:

Creates an application VD within the device driver and thus a connection to a device. In the case of success, the function sets the transition of the created VD instance into the state Preparation.

In the case of no errors the reference of the data object, which describes the input parameter of GDI_Initiate, is not valid any longer.

Input Parameters:

| | |
|---------------------------|---|
| GDICOMMTYPE Communication | type of communication |
| INSTANCEITEM InstanceName | name of the created application VD instance |
| GDIDataBaseType* pDataObj | reference of the data object, which describes the input parameter of GDI_Initiate |

Output Parameters:

| | |
|--|--|
| GDIDeviceConnection*& pGDIDeviceConnection | reference of the object, which represents a created device connection – the reference is only valid if the call was executed faultless |
| GDIERROR* pExecutionErr | error message of the GDI-API function |

Return value:

Error message of the GDI coordinator

3.5 INTERFACE DCDDDEVICEFUNCTION

Basis class

DCDBaseltem

Task

The interface contains informations about the selected device function, which is defined by the selected dcd file.

3.5.1 GDICOGETDCDCOMOBJITERATOR

Prototype:

```
short GDICOGetDCDComObjlterator ( DCDCComObjectlterator*&  
                                   pComObjlterator)
```

Task:

The function gets a reference to an iterator, which is used for the output of all comobjects of the device function, which are defined by the dcd file.

Input Parameters:

Output Parameters:

| | |
|--|--|
| DCDCComObjectlterator*& pGDIComObjlterator | reference of the comobject iterator |
|--|--|

Return value:

Error message of the GDI coordinator

3.5.2 GDICOGETDCDOPERATIONITERATOR

Prototype:

```
short GDICOGetDCDOperationIterator (DCDOperationIterator*&  
                                     pOperationIterator)
```

Task:

The function gets a reference to an iterator, which is used for the output of all operations of the device function, which are defined by the dcd file.

Input Parameters:**Output Parameters:**

| | |
|--|-------------------------------------|
| DCDOperationIterator*& pGDIOperationIterator | reference of the operation iterator |
|--|-------------------------------------|

Return value:

Error message of the GDI coordinator

3.6 INTERFACE DCDCOMOBJECT

Basis class

DCDBaseltem

Task

The interface contains informations about the appropriate communication object, which is defined by the selected dcd file.

3.6.1 GDICOGETPROPERTIES

Prototype:

```
short GDICOGetProperties (    COMOBJECTTYPE* pType,  
                             GDIACCRIGHTS* pAccessRights)
```

Task:

The function gets the properties of the comobject.

Input Parameters:

Output Parameters:

| | |
|-----------------------------|---|
| COMOBJECTTYPE* pType | type of the communication object |
| GDIACCRIGHTS* pAccessRights | access rights of the communication object |

Return value:

Error message of the GDI coordinator

3.7 INTERFACE DCDOPERATION

Basis class

DCDBaseltem

Task

The interface contains informations about the appropriate operation, which is defined by the selected dcd file.

3.8 INTERFACE **GDIDEVICECONNECTION**

Basis class

DCDBaseltem

Task

The interface supports the control of a created VD instance.

3.8.1 **GDICOGETINSTANCENAME**

Prototype:

```
short GDICOGetInstanceName ( INSTANCEITEM* pInstanceName)
```

Task:

The function gets the Name of the created VD instance.

Input Parameters:

Output Parameters:

| | |
|-----------------------------|---------------|
| INSTANCEITEM* pInstanceName | instance name |
|-----------------------------|---------------|

Return value:

Error message of the GDI coordinator

3.8.2 GDICOVDIDENTIFY

Prototype:

```
short GDICOVDIdentify (  GDIIDENT* pIdentInformation,  
                        GDIERROR* pExecutionErr)
```

Task:

The function calls the GDI driver function *GDI_Identify*.
GDI_Identify delivers data for identification of the device manufacturers and the connected devices (VDs).

Input Parameters:**Output Parameters:**

GDIIDENT* pIdentInformation Ident Information

GDIERROR* pExecutionErr error message of the GDI-API function

Return value:

Error message of the GDI coordinator

3.8.3 GDICOVDSTATUS

Prototype:

```
short GDICOVDStatus (  GDISTAT* pStatusInformation,  
                      GDIERROR* pExecutionErr)
```

Task:

The function calls the GDI driver function *GDI_Status*.
GDI_Status delivers the current status of the VD and the connected device respectively.

Input Parameters:**Output Parameters:**

| | |
|-----------------------------|---------------------------------------|
| GDISTAT* pStatusInformation | Information of the VD |
| GDIERROR* pExecutionErr | error message of the GDI-API function |

Return value:

Error message of the GDI coordinator

3.8.5 GDICOSTOPWORKING

Prototype:

short GDICOSTopWorking (GDICOMMTYPE Communication,
GDIERROR* pExecutionErr)

Task:

The function switches the VD instance into the *Evaluationphase*.
The function switches in the following GDI states:
Working -> *Evaluation*

Input Parameters:

GDICOMMTYPE Communication type of communication

Output Parameters:

GDIERROR* pExecutionErr error message of the GDI-API function

Return value:

Error message of the GDI coordinator

3.8.6 GDICOCREATEDEVICEFUNCTION

Prototype:

```
short GDICOCreateDeviceFunction (    const char* DevFuncName,
                                     const INSTANCEITEM InstanceName,
                                     GDIDataBaseType* pDataObj,
                                     GDICOMMTYPE Communication,
                                     GDIDeviceFunction*& pGDIDeviceFunction,
                                     GDIERROR* pExecutionErr)
```

Task:

The function creates a instance of a device function (Func Object).

For this *GDI_CreateDeviceFunction* is executed, after the respective VD handle was transferred into the status *Preparation* or *Revise*.

Before the call of the function always the call of *GDICOGetCreateParameterData* is necessary for the transfer of the create parameters. If *GDICOGetCreateParameterData* was not called before the execution by *GDICOCreateDeviceFunction*, then an error message of the coordinator takes place.

The data object of the input parameter of the function object, reserved by *GDICOGetCreateParameterData*, is no longer valid in each case after the call of *GDICOCreateDeviceFunction* and cannot be used therefore again for the production of a function object of same type.

In the case of no errors the reference of the data object, which describes the create parameter of the device function, is not valid any longer.

Input Parameters:

| | |
|---------------------------|---|
| char* DevFuncName | Class name of the device function |
| INSTANCEITEM InstanceName | instance name |
| GDIDataBaseType* pDataObj | reference of the data object, which describes the create parameter of the device function |
| GDICOMMTYPE Communication | type of communication |

Output Parameters:

| | |
|--|---|
| GDIDeviceFunction*& pGDIDeviceFunction | reference of the func object – the reference is only valid if the call was executed faultless |
| GDIERROR* pExecutionErr | error message of the GDI-API function |

Return value:



Error message of the GDI coordinator

3.8.7 GDICOGETCREATEPARAMETERDATA

Prototype:

```
short GDICOGetCreateParameterData (  const char* FuncObjName,  
                                     GDIDataBaseType*& pDataObj)
```

Task:

The function supplies the reference to a data object, which contains the CreateParameter of the device function.

Input Parameters:

| | |
|-------------------|-----------------------------------|
| char* FuncObjName | Class name of the function object |
|-------------------|-----------------------------------|

Output Parameters:

| | |
|----------------------------|---|
| GDIDataBaseType*& pDataObj | reference of the data object, which describes the create parameter of the device function |
|----------------------------|---|

Return value:

Error message of the GDI coordinator

3.8.8 GDICODELETEDEVICEFUNCTION

Prototype:

```
short GDICODEleteDeviceFunction (    GDICOMMTYPE Communication,  
                                     GDIDeviceFunction* pGDIDeviceFunction,  
                                     GDIERROR* pExecutionErr)
```

Task:

The function deletes a device function.

The call of this function releases the execution of *GDI_DeleteFunctionObject* for the appropriate device function.

For this the respective VD instance is transferred into the evaluation status and all member functions of the device function with *GDI_DeleteCommObject* are closed. Furthermore all data areas of the device function created by the coordinator and not eliminated yet are deleted.

Input Parameters:

| | |
|---------------------------------------|-----------------------------|
| GDIDeviceFunction* pGDIDeviceFunction | reference of the funcobject |
| GDICOMMTYPE Communication | type of communication |

Output Parameters:

| | |
|-------------------------|---------------------------------------|
| GDIERROR* pExecutionErr | error message of the GDI-API function |
|-------------------------|---------------------------------------|

Return value:

Error message of the GDI coordinator

3.9 INTERFACE GDIDeviceFunction

Basis class

DCDDeviceFunction

Task

The interface supports the control of a created instance of a device function.

3.9.1 GDICoGetInstanceName

Prototype:

```
short GDICoGetInstanceName ( INSTANCEITEM* pInstanceName)
```

Task:

The function gets the Name of the created Function object instance.

Input Parameters:

Output Parameters:

| | |
|-----------------------------|---------------|
| INSTANCEITEM* pInstanceName | instance name |
|-----------------------------|---------------|

Return value:

Error message of the GDI coordinator

3.9.2 GDICOCREATECOMOBJECT

Prototype:

```
short GDICOCreateComObject ( const char* ComObjName,  
                             INSTANCEITEM InstanceName,  
                             GDICOMMTYPE Communication;  
                             GDIComObject*& pGDIComObject,  
                             GDIERROR* pExecutionErr)
```

Task:

The creates a instance of an communication object.
For this purpose the VD of the appropriate VD instance is brought either into the *Preparation* or into the *Revisephase*. With communication objects takes place additionally the call of *GDI_CreateCommObject*.
If default data are available in the DCD, the respective input parameter with these is initialized.

Input Parameters:

| | |
|---------------------------|---------------------------|
| char* ComObjName | communication object name |
| INSTANCEITEM InstanceName | instance name |
| GDICOMMTYPE Communication | type of communication |

Output Parameters:

| | |
|------------------------------|--|
| GDIComObject*& pGDIComObject | reference of the communication object – the reference is only valid if the call was executed faultless |
| GDIERROR* pExecutionErr | error message of the GDI-API function |

Return value:

Error message of the GDI coordinator

3.9.3 GDICOCREATEOPERATION

Prototype:

```
short GDICOCreateOperation (  const char* OperationName,  
                              INSTANCEITEM InstanceName,  
                              GDIOperation*& pGDIOperation,  
                              GDIERROR* pExecutionErr)
```

Task:

The creates a instance of an operation.
If default data are available in the DCD, the respective input parameter with these is initialized.

Input Parameters:

| | |
|---------------------------|-----------------------|
| char* OperationName | name of the operation |
| INSTANCEITEM InstanceName | instance name |

Output Parameters:

| | |
|------------------------------|---|
| GDIOperation*& pGDIOperation | reference of the operation – the reference is only valid if the call was executed faultless |
| GDIERROR* pExecutionErr | error message of the GDI-API function |

Return value:

Error message of the GDI coordinator

3.9.4 GDICOGETCOMOBJECTBYNAME

Prototype:

```
short GDICOGetComObjectByName (  const char* ComObjName,  
                                GDIComObject*& pGDIComObject)
```

Task:

The function gets a reference to an existing communication object.

Input Parameters:

| | |
|------------------|---------------------------------|
| char* ComObjName | Identifier of expected Instance |
|------------------|---------------------------------|

Output Parameters:

| | |
|------------------------------|---|
| GDIComObject*& pGDIComObject | reference of the instance of the communication object |
|------------------------------|---|

Return value:

Error message of the GDI coordinator

3.9.5 GDICOGETCOMOBJECTITERATOR

Prototype:

short GDICOGetComObjectlterator (GDIComObjectlterator*& pGDIComObjectlterator)

Task:

The function gets a reference to an iterator, which is used for the output of all set up communication objects of the device function.

Input Parameters:**Output Parameters:**

| | |
|--|--|
| GDIComObjectlterator*& pGDIComObjectlterator | reference of the communication object iterator |
|--|--|

Return value:

Error message of the GDI coordinator

3.9.6 GDICOGETOPERATIONBYNAME

Prototype:

```
short GDICOGetComObjectByName (  const char* OperationName,  
                                GDIOperation*& pGDIOperation)
```

Task:

The function gets a reference to an existing operation instance.

Input Parameters:

char* OperationName Identifier of expected Instance

Output Parameters:

GDIOperation*& pGDIOperation reference of the instance of the
operation

Return value:

Error message of the GDI coordinator

3.9.7 GDICOGETOPERATIONITERATOR

Prototype:

```
short GDICOGetComOperationIterator ( GDIOperationIterator*&  
                                     pGDIOperationIterator)
```

Task:

The function gets a reference to an iterator, which is used for the output of all set up operation of the device function.

Input Parameters:**Output Parameters:**

| | |
|--|-------------------------------------|
| GDIOperationIterator*& pGDIOperationIterator | reference of the operation iterator |
|--|-------------------------------------|

Return value:

Error message of the GDI coordinator

3.10 INTERFACE GDICOMOBJECT

Basis class

DCDComObject

Task

The interface supports the control of a created instance of a communication object.

3.10.1 GDICOGETINSTANCENAME

Prototype:

```
short GDICOGetInstanceName ( INSTANCEITEM* pInstanceName)
```

Task:

The function gets the Name of the created communication object instance.

Input Parameters:

Output Parameters:

INSTANCEITEM* pInstanceName instance name

Return value:

Error message of the GDI coordinator

3.10.2 GDICOGETINPUTPARAMETERDATA

Prototype:

short GDICOGetInputParameterData (GDIDataBaseType*& pDataObj)

Task:

The function gets reference of the input data object.

Input Parameters:**Output Parameters:**

| | |
|----------------------------|---|
| GDIDataBaseType*& pDataObj | reference of the data object, which describes the input parameter of the communication object |
|----------------------------|---|

Return value:

Error message of the GDI coordinator

3.10.3 GDICOGETOUTPUTPARAMETERDATA

Prototype:

short GDICOGetOutputParameterData (GDIDataBaseType*& pDataObj)

Task:

The function gets reference of the output data object.

Input Parameters:**Output Parameters:**

| | |
|----------------------------|--|
| GDIDataBaseType*& pDataObj | reference of the data object, which describes the output parameter of the communication object |
|----------------------------|--|

Return value:

Error message of the GDI coordinator

3.10.4 GDICOWRITE

Prototype:

short GDICOWrite (GDICOMMTYPE Communication ,GDIERROR* pExecutionErr)

Task:

The function writes the input parameter of the communication object. The coordinator executes for this purpose a *GDI_Write* with the appropriate communication object. For this the respective VD is brought either into the *Preparation-*, *Revise-* or into the *Working-phase*.

The coordinator uses the values of the input data object instance for the exchange of data.

Input Parameters:

GDICOMMTYPE Communication type of communication

Output Parameters:

GDIERROR* pExecutionErr error message of the GDI-API function

Return value:

Error message of the GDI coordinator

3.10.5 GDICOREAD

Prototype:

short GDICORead (GDICOMMTYPE Communication ,GDIERROR* pExecutionErr)

Task:

The function reads the output parameter of the communication object. The coordinator executes for this purpose a *GDI_Read* on the appropriate communication object. For this the respective VD is brought either into the *Preparation-*, *Revise-* or into the *Working-phase*.

The output values remain within the area of the coordinator. The coordinator uses the values of the output data object instance for the exchange of data.

Input Parameters:

GDICOMMTYPE Communication type of communication

Output Parameters:

GDIERROR* pExecutionErr error message of the GDI-API function

Return value:

Error message of the GDI coordinator

3.10.6 GDICOACCEPTOCCURED

Prototype:

short GDICOAcceptOccured (ACCEPT* pAccOccured)

Task:

Returns the information, if an Accept has been carried out since the last test or GDICOWrite. The service is used, if no call back is supported by the application.

Input Parameters:**Output Parameters:**

ACCEPT* pAccOccured informations about occurred GDI_Accept

Return value:

Error message of the GDI coordinator

3.10.7 GDICOINFREPORTOCCURED

Prototype:

short GDICOInfReportOccured (INFREPORT* pInfRepOccured)

Task:

Returns the information, if an Information Report has been carried out since the last test or `gdiReadValue`. The service is used, if no call back is supported by the application.

Input Parameters:**Output Parameters:**

| | |
|---------------------------|--|
| INFREPORT* pInfRepOccured | informations about occurred GDI_InformationReport |
|---------------------------|--|

Return value:

Error message of the GDI coordinator

3.11 INTERFACE GDIOPERATION

Basis class

DCDOperation

Task

The interface supports the control of a created instance of a operation.

3.11.1 GDICOGETINSTANCENAME

Prototype:

```
short GDICOGetInstanceName ( INSTANCEITEM* pInstanceName)
```

Task:

The function gets the Name of the created communication object instance.

Input Parameters:

Output Parameters:

| | |
|-----------------------------|---------------|
| INSTANCEITEM* pInstanceName | instance name |
|-----------------------------|---------------|

Return value:

Error message of the GDI coordinator

3.11.2 GDICOGETINPUTPARAMETERDATA

Prototype:

short GDICOGetInputParameterData (GDIDataBaseType*& pDataObj)

Task:

The function gets reference of the input data object.

Input Parameters:**Output Parameters:**

| | |
|----------------------------|--|
| GDIDataBaseType*& pDataObj | reference of the data object, which describes the input parameter of the operation |
|----------------------------|--|

Return value:

Error message of the GDI coordinator

3.11.3 GDICOGETOUTPUTPARAMETERDATA

Prototype:

short GDICOGetOutputParameterData (GDIDataBaseType*& pDataObj)

Task:

The function gets reference of the output data object.

Input Parameters:**Output Parameters:**

| | |
|----------------------------|---|
| GDIDataBaseType*& pDataObj | reference of the data object, which describes the output parameter of the operation |
|----------------------------|---|

Return value:

Error message of the GDI coordinator

3.12 CALL BACK FUNCTIONS

Task

The call back functions are called by the coordinator. The application defines the appropriate functions. The publication of the this functions takes place via the function *GDICOCreateGDIWorkSpace*.

3.12.1 GDIMSGTRANSFERCOMPLETECO

Prototype:

```
void GDIMSGTransferCompleteCO (      GDIComObject* pGDIComObject,  
                                  GDIERROR ExecutionErr)
```

Task:

The function announces the end of an asynchronous transfer between coordinator and GDI driver or between GDI driver and device to the superordinate program. The function is called by the coordinator, if the GDI driver calls the function *GDI_Complete* of the coordinator.

Input Parameters:

| | |
|------------------------------|---|
| GDIComObject RefGDIComObject | reference of the instance of the communication object |
| GDIERROR ExecutionErr | error message of the GDI-API function |

Output Parameters:

3.12.2 GDIMSGTRANSFERCOMPLETEOP

Prototype:

```
void GDIMSGTransferCompleteOp (      GDIOperation* pGDIOperation,  
                                   GDIErr ExecutionErr)
```

Task:

The function announces the end of an asynchronous transfer between coordinator and GDI driver or between GDI driver and device to the superordinate program. The function is called by the coordinator, if the GDI driver calls the function *GDI_Complete* of the coordinator.

Input Parameters:

| | |
|------------------------------|--|
| GDIOperation RefGDIOperation | reference of the instance of the operation |
| GDIErr ExecutionErr | error message of the GDI-API function |

Output Parameters:

3.12.3 GDIMSGDRIVERREPORT

Prototype:

```
void GDIMSGDriverReport (GDIComObject* pGDIComObject,  
                        GDIDataBaseType* pDataObj)
```

Task:

The function announces a *GDI_InformationReport* call of the driver to the application. The results of the transmitted dataobject (*DataObj*) are valid only within the function *GDIMSGDriverReport*.

Input Parameters:

| | |
|-----------------------------|--|
| GDIComObject* pGDIComObject | reference of the instance of the communication object |
| GDIDataBaseType* pDataObj | reference of the data object, which describes the output parameter of the <i>GDI_InformationReport</i> |

Output Parameters:

3.12.4 GDIMSGEXCEPTION

Prototype:

```
void GDIMSGException (GDIWorkspace* pGDIWorkspaceObj, short ErrorCode)
```

Task:

The function announces an error exception.

Input Parameters:

| | |
|--------------------------------|--|
| GDIWorkspace* pGDIWorkspaceObj | reference of the GDI Work Space Object |
| unsigned short Errorcode | error code of the exception |

Output Parameters:

3.12.5 GDIMSGACCEPT

Prototype:

```
void GDIMSGAccept (    GDIComObject* pGDIComObject,  
                     GDIDataBaseType* pDataObj)
```

Task:

The function announces a *GDI_Accept* call of the driver to the application.

Input Parameters:

| | |
|-----------------------------|--|
| GDIComObject* pGDIComObject | reference of the instance of the communication object |
| GDIDataBaseType* pDataObj | reference of the data object, which describes the input parameter of the <i>GDI_Accept</i> |

Output Parameters:

3.12.6 GDIMSGCREATEDEVICECONNECTION

Prototype:

```
void GDIMSGCreateDeviceConnection (   GDIWorkSpace* pGDIWorkSpaceObj,  
                                     GDIERROR ExecutionErr);
```

Task:

The function announces the end of an *GDICOCreateDeviceConnection* call. The function is called by the coordinator, if the GDI driver calls the function *GDI_Complete* of the coordinator.

Input Parameters:

| | |
|--------------------------------|--|
| GDIWorkSpace* pGDIWorkSpaceObj | reference of the GDI Work Space Object |
| GDIERROR ExecutionErr | error message of the GDI-API function |

Output Parameters:

3.12.7 GDIMSGCREATEDeviceFUNCTION

Prototype:

```
void GDIMSGCreateDeviceFunction (    GDIWorkSpace* pGDIWorkSpaceObj,  
                                     GDIERROR ExecutionErr);
```

Task:

The function announces the end of a *GDICOCreateDeviceFunction* call. The function is called by the coordinator, if the GDI driver calls the function *GDI_Complete* of the coordinator.

Input Parameters:

| | |
|--------------------------------|--|
| GDIWorkSpace* pGDIWorkSpaceObj | reference of the GDI Work Space Object |
| GDIERROR ExecutionErr | error message of the GDI-API function |

Output Parameters:

3.12.8 GDIMSGCREATECOMOBJECT

Prototype:

```
void GDIMSGCreateComObject (  GDIWorkSpace* pGDIWorkSpaceObj,  
                             GDIERROR ExecutionErr);
```

Task:

The function announces the end of a *GCICOCreateComObject* call. The function is called by the coordinator, if the GDI driver calls the function *GDI_Complete* of the coordinator.

Input Parameters:

| | |
|--------------------------------|--|
| GDIWorkSpace* pGDIWorkSpaceObj | reference of the GDI Work Space Object |
| GDIERROR ExecutionErr | error message of the GDI-API function |

Output Parameters:

3.13 INTERFACE GDIDATABASETYPE

Basis class

DCDBasItem

Task

The GDI server reserves memory for the input and output parameters of all functions of the gdi driver, which are defined in the dcd file. An object of the type GDIDataBaseType administers this created memory as well as the attributes of the appropriate input or output parameter. Reading or modifying from data is always made by the data object belonging to the parameter.

3.13.1 GDIDATAGETYPE

Prototype:

GDIDATATYPE GDIDATAGetType()

Task:

The function supplies information about the data type.

Input Parameters:

Output Parameters:

Return value:

GDIDATATYPE Type as numeric value

3.13.2 GDIDATAGETINSTANCEVERSION

Prototype:

DATAINSTANCEVERSION GDIDATAGetInstanceVersion ()

Task:

Returns the description of the derivation of the datum.

Input Parameters:**Output Parameters:****Return value:**

| | |
|---------------------|---|
| DATAINSTANCEVERSION | description of the derivation of the datum. |
|---------------------|---|

3.13.3 GDIDATAGETORDVALUE

Prototype:

unsigned long GDIDATAGetOrdValue()

Task:

Returns the internal position value of the type (sub type) in a structure or union. Is used for the detection of the switch value of a sub type within a union. Within a structure this value is used for information only. If the type is a sub type in sequences or arrays, this service returns the value 1.

Input Parameters:**Output Parameters:****Return value:**

unsigned long Internal position of this type as sub type within a structure or union.

3.13.4 GDIDATASETVALUE

Prototype:

short GDIDATASETVALUE (BYTE* pDataValue)

Task:

Sets the value of a data area within the coordinator with the value handed over by the reference. The data are handled as a block of Bytes. The number of Bytes is determined by the type of the datum.

If the type of the data object a sequence, the function copies the data of *pDataValue* into the data area of the sequence. The header of the sequence is not changed. The size of *pDataValue* must be the same like the current size of the sequence. Before the function can be used, the current size of the sequence must be set by the function *GDIDATASETCURRENTSIZE*.

Note: *GDIDATASETCURRENTSIZE* sets only the number of elements of a sequence. The coordinator calculates the size of the data area in bytes.

If the type of the data object a structure or a union and a sequence is a member of this, the function sets only the header structure of the sequence. The data area of the sequence is not changed. In this case it is necessary the application looks for the data object of this sequence in the structure/union.

If the data object a selector of a union then it is not possible to use this function. (starting from Version 1.1.4.0.)

Input Parameters:

BYTE* pDataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.13.5 GDIDATAGETVALUE

Prototype:

short GDIDATAGetValue (BYTE* pDataValue)

Task:

Returns the value of the internal data area of the coordinator. The number of Bytes is determined by the type of the datum.

If the type of the data object a sequence, the function copies the data from the data area of the sequence into *pDataValue* . The function delivers not the header of the sequence. The size of *pDataValue* must be the same like the current size of the sequence. Before the function can be used, the current size of the sequence must be read by the function *GDIDATAGetCurrentSize*.

Note: *GDIDATAGetCurrentSize* gets only the number of elements of a sequence. The application calculates the size of *pDataValue* in bytes.

If the type of the data object a structure or a union and a sequence is a member of this, the function gets only the header structure of the sequence. In this case it is necessary the application looks for the data object of this sequence in the structure/union.

Input Parameters:**Output Parameters:**

BYTE* pDataValue value of a data area

Return value:

Error message of the GDI coordinator

3.14 INTERFACE GDIDATASHORT

Basis class

GDIDataBaseType

3.14.1 GDIDATAGETMAXVALUE

Prototype:

short GDIDATAGetMaxValue ()

Task:

Returns the maximum value of a datum, based upon this type.

Input Parameters:

Output Parameters:

Return value:

Maximum value of a datum, based upon this type.

3.14.2 GDIDATAGETMINVALUE

Prototype:

short GDIDATAGetMinValue ()

Task:

Returns the minimum value of a datum, based upon this type.

Input Parameters:**Output Parameters:****Return value:**

Minimum value of a datum, based upon this type.

3.14.3 GDIDATAGETSTEPVALUE

Prototype:

short GDIDATAGetStepValue ()

Task:

Returns the step width of a type, which defines valid elements of a datum starting from the minimum value. If the step value is not defined explicitly, it takes the value 1.

Input Parameters:**Output Parameters:****Return value:**

Step width of a datum, based upon this type.

3.14.4 GDIDATASETSHORTVALUE

Prototype:

short GDIDATASetShortValue (short DataValue)

Task:

Copies a short value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

short DataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.14.5 GDIDATAGETSHORTVALUE

Prototype:

short GDIDATAGetShortValue ()

Task:

Returns the value of a short datum. This might also be a sub-element of the type short of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of the data area.

3.15 INTERFACE GDIDATAUSHORT

Basis class

GDIDataBaseType

3.15.1 GDIDATAGETMAXVALUE

Prototype:

unsigned short GDIDATAGetMaxValue ()

Task:

Returns the maximum value of a datum, based upon this type.

Input Parameters:

Output Parameters:

Return value:

Maximum value of a datum, based upon this type.

3.15.2 GDIDATAGETMINVALUE

Prototype:

unsigned short GDIDATAGetMinValue ()

Task:

Returns the minimum value of a datum, based upon this type.

Input Parameters:**Output Parameters:****Return value:**

Minimum value of a datum, based upon this type.

3.15.3 GDIDATAGETSTEPVALUE

Prototype:

unsigned short GDIDATAGetStepValue ()

Task:

Returns the step width of a type, which defines valid elements of a datum starting from the minimum value. If the step value is not defined explicitly, it takes the value 1.

Input Parameters:**Output Parameters:****Return value:**

Step width of a datum, based upon this type.

3.15.4 GDIDATASETUSHORTVALUE

Prototype:

short GDIDATASetUShortValue (unsigned short DataValue)

Task:

Copies a unsigned short value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

unsigned short DataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.15.5 GDIDATAGETUSHORTVALUE

Prototype:

unsigned short GDIDATAGetUShortValue ()

Task:

Returns the value of a unsigned short datum. This might also be a sub-element of the type unsigned short of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of a data area.

3.16 INTERFACE GDIDATAULONG

Basis class

GDIDataBaseType

3.16.1 GDIDATAGETMAXVALUE

Prototype:

unsigned long GDIDATAGetMaxValue()

Task:

Returns the maximum value of a datum, based upon this type.

Input Parameters:

Output Parameters:

Return value:

Maximum value of a datum, based upon this type.

3.16.2 GDIDATAGETMINVALUE

Prototype:

unsigned long GDIDATAGetMinValue ()

Task:

Returns the minimum value of a datum, based upon this type.

Input Parameters:**Output Parameters:****Return value:**

Minimum value of a datum, based upon this type.

3.16.3 GDIDATAGETSTEPVALUE

Prototype:

unsigned long GDIDATAGetStepValue ()

Task:

Returns the step width of a type, which defines valid elements of a datum starting from the minimum value. If the step value is not defined explicitly, it takes the value 1.

Input Parameters:**Output Parameters:****Return value:**

Step width of a datum, based upon this type.

3.16.4 GDIDATASETULONGVALUE

Prototype:

short GDIDATASETULONGVALUE (unsigned long DataValue)

Task:

Copies a unsigned long value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

unsigned long DataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.16.5 GDIDATAGETULONGVALUE

Prototype:

unsigned long GDIDATAGetULongValue ()

Task:

Returns the value of a unsigned long datum. This might also be a sub-element of the type unsigned long of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of a data area

3.17 INTERFACE GDIDATALONG

Basis class

GDIDataBaseType

3.17.1 GDIDATAGETMAXVALUE

Prototype:

long GDIDATAGetMaxValue ()

Task:

Returns the maximum value of a datum, based upon this type.

Input Parameters:

Output Parameters:

Return value:

Maximum value of a datum, based upon this type.

3.17.2 GDIDATAGETMINVALUE

Prototype:

long GDIDATAGetMinValue ()

Task:

Returns the minimum value of a datum, based upon this type.

Input Parameters:**Output Parameters:****Return value:**

Minimum value of a datum, based upon this type.

3.17.3 GDIDATAGETSTEPVALUE

Prototype:

long GDIDATAGetStepValue ()

Task:

Returns the step width of a type, which defines valid elements of a datum starting from the minimum value. If the step value is not defined explicitly, it takes the value 1.

Input Parameters:**Output Parameters:****Return value:**

Step width of a datum, based upon this type.

3.17.4 GDIDATASETLONGVALUE

Prototype:

short GDIDATASetLongValue (long DataValue)

Task:

Copies a long value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

long DataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.17.5 GDIDATAGETLONGVALUE

Prototype:

long GDIDATAGetLongValue ()

Task:

Returns the value of a long datum. This might also be a sub-element of the type unsigned long of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of a data area.

3.18 INTERFACE GDIDATAFLOAT

Basis class

GDIDataBaseType

3.18.1 GDIDATAGETMAXVALUE

Prototype:

float GDIDATAGetMaxValue ()

Task:

Returns the maximum value of a datum, based upon this type.

Input Parameters:

Output Parameters:

Return value:

Maximum value of a datum, based upon this type.

3.18.2 GDIDATAGETMINVALUE

Prototype:

float GDIDATAGetMinValue ()

Task:

Returns the minimum value of a datum, based upon this type.

Input Parameters:**Output Parameters:****Return value:**

Minimum value of a datum, based upon this type.

3.18.3 GDIDATAGETSTEPVALUE

Prototype:

float GDIDATAGetStepValue ()

Task:

Returns the step width of a type, which defines valid elements of a datum starting from the minimum value. If the step value is not defined explicitly, it takes the value 1.

Input Parameters:**Output Parameters:****Return value:**

Step width of a datum, based upon this type.

3.18.4 GDIDATASETFloatVALUE

Prototype:

short GDIDATASetFloatValue (float DataValue)

Task:

Copies a float value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

float DataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.18.5 GDIDATAGETFLOATVALUE

Prototype:

float GDIDATAGetFloatValue ()

Task:

Returns the value of a float datum. This might also be a sub-element of the type unsigned float of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of a data area.

3.19 INTERFACE GDIDATADouble

Basis class

GDIDataBaseType

3.19.1 GDIDATAGetMaxValue

Prototype:

double GDIDATAGetMaxValue ()

Task:

Returns the maximum value of a datum, based upon this type.

Input Parameters:

Output Parameters:

Return value:

Maximum value of a datum, based upon this type.

3.19.2 GDIDATAGETMINVALUE

Prototype:

double GDIDATAGetMinValue ()

Task:

Returns the minimum value of a datum, based upon this type.

Input Parameters:**Output Parameters:****Return value:**

Minimum value of a datum, based upon this type.

3.19.3 GDIDATAGETSTEPVALUE

Prototype:

```
double GDIDATAGetStepValue ()
```

Task:

Returns the step width of a type, which defines valid elements of a datum starting from the minimum value. If the step value is not defined explicitly, it takes the value 1.

Input Parameters:**Output Parameters:****Return value:**

Step width of a datum, based upon this type.

3.19.4 GDIDATASETDOUBLEVALUE

Prototype:

short GDIDATASetDoubleValue (double DataValue)

Task:

Copies a double value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

double DataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.19.5 GDIDATAGETDOUBLEVALUE

Prototype:

double GDIDATAGetDoubleValue ()

Task:

Returns the value of a double datum. This might also be a sub-element of the type unsigned double of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of a data area.

3.20 INTERFACE GDIDATACHAR

Basis class

GDIDataBaseType

3.20.1 GDIDATAGETMAXVALUE

Prototype:

```
char GDIDATAGetMaxValue ()
```

Task:

Returns the maximum value of a datum, based upon this type.

Input Parameters:

Output Parameters:

Return value:

Maximum value of a datum, based upon this type.

3.20.2 GDIDATAGETMINVALUE

Prototype:

char GDIDATAGetMinValue ()

Task:

Returns the minimum value of a datum, based upon this type.

Input Parameters:**Output Parameters:****Return value:**

Minimum value of a datum, based upon this type.

3.20.3 GDIDATAGETSTEPVALUE

Prototype:

```
char GDIDATAGetStepValue ()
```

Task:

Returns the step width of a type, which defines valid elements of a datum starting from the minimum value. If the step value is not defined explicitly, it takes the value 1.

Input Parameters:

Output Parameters:

Return value:

Step width of a datum, based upon this type.

3.20.4 GDIDATASETCHARVALUE

Prototype:

short GDIDATASetCharValue (char DataValue)

Task:

Copies a char value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

char DataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.20.5 GDIDATAGETCHARVALUE

Prototype:

```
char GDIDATAGetCharValue ()
```

Task:

Returns the value of a char datum. This might also be a sub-element of the type unsigned char of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of a data area.

3.21 INTERFACE GDIDATAUCHAR

Basis class

GDIDataBaseType

3.21.1 GDIDATAGETMAXVALUE

Prototype:

unsigned char GDIDATAGetMaxValue ()

Task:

Returns the maximum value of a datum, based upon this type.

Input Parameters:

Output Parameters:

Return value:

Maximum value of a datum, based upon this type.

3.21.2 GDIDATAGETMINVALUE

Prototype:

unsigned char GDIDATAGetMinValue ()

Task:

Returns the minimum value of a datum, based upon this type.

Input Parameters:**Output Parameters:****Return value:**

Minimum value of a datum, based upon this type.

3.21.3 GDIDATAGETSTEPVALUE

Prototype:

unsigned char GDIDATAGetStepValue ()

Task:

Returns the step width of a type, which defines valid elements of a datum starting from the minimum value. If the step value is not defined explicitly, it takes the value 1.

Input Parameters:**Output Parameters:****Return value:**

Step width of a datum, based upon this type.

3.21.4 GDIDATASETCHARVALUE

Prototype:

short GDIDATASetUCharValue (unsigned char DataValue)

Task:

Copies a char value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

unsigned char DataValue value of a data area

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.21.5 GDIDATAGETCHARVALUE

Prototype:

unsigned char GDIDATAGetUCharValue ()

Task:

Returns the value of a char datum. This might also be a sub-element of the type unsigned char of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of a data area.

3.22 INTERFACE GDIDATABOOLEAN

Basis class

GDIDataBaseType

3.22.1 GDIDATASETBOOLEANVALUE

Prototype:

short GDIDATASetBooleanValue (boolean DataValue)

Task:

Copies a boolean value to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

boolean DataValue value of a data area

Output Parameters:

Return value:

Error message of the GDI coordinator

3.22.2 GDIDATAGETBOOLEANVALUE

Prototype:

boolean GDIDATAGetBooleanValue ()

Task:

Returns the value of a boolean datum. This might also be a sub-element of the type unsigned boolean of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

Value of a data area.

3.23 INTERFACE GDIDATADEVFUNCREFERENCE

Basis class

GDIDataBaseType

3.23.1 GDIDATASETDEVFUNC

Prototype:

```
short GDIDATASetDevFunc(GDIDeviceFunction* pDeviceFunction)
```

Task:

Copies the handle and the id of the function object instance to the data area of the coordinator. This service is also used to fill sub areas of a datum with values (elements of Arrays, Sequences, Structures, Unions).

Input Parameters:

| | |
|--------------------|-----------------|
| GDIDeviceFunction* | pDeviceFunction |
|--------------------|-----------------|

Output Parameters:

Return value:

Error message of the GDI coordinator

3.23.2 GDIDATAGETDEVFUNC

Prototype:

GDIDeviceFunction* GDIDATAGetDevFunc()

Task:

Returns reference of a func object. This might also be a sub-element of the type unsigned boolean of a datum (Array, Sequence, Structure, Union).

Input Parameters:**Output Parameters:****Return value:**

reference of a func object

3.24 INTERFACE GDIDATAENUM

Basis class

GDIDataShort

3.24.1 GDIDATAGETFIRSTELEMENT

Prototype:

short GDIDATAGetFirstElement (GDIENUMERATIONMEMBER* pMember)

Task:

Returns the first element of an enumeration of single elements. Usually this is the element with the lowest value. The return element is a structure which consists of the sub elements String and short.

Input Parameters:

Output Parameters:

| | |
|-------------------------------|--|
| GDIENUMERATIONMEMBER* pMember | Structure consisting of the sub elements String and short. If no element is available, the value „noGDIElement“ as name and the numeric value 32768 is returned. |
|-------------------------------|--|

Return value:

Error message of the GDI coordinator

3.24.2 GDIDATAGETNEXTELEMENT

Prototype:

short GDIDATAGetNextElement (GDIENUMERATIONMEMBER* pMember)

Task:

Returns the next element of an enumeration of single elements. The return element is a structure which consists of the sub elements String and short.

Input Parameters:**Output Parameters:**

| | |
|-------------------------------|--|
| GDIENUMERATIONMEMBER* pMember | Structure consisting of the sub elements String and short. If no element is available, the value „noGDIElement“ as name and the numeric value 32768 is returned. |
|-------------------------------|--|

Return value:

Error message of the GDI coordinator

3.25 INTERFACE GDIDATASTRUCTURE

Basis class

GDIDataBaseType

3.25.1 GDIDATAGETDATATYPEITERATOR

Prototype:

```
short GDIDATAGetDataTypeliterator ( GDIDataTypeliterator*&  
                                     pGDIDataTypeliterator)
```

Task:

Generates a type iterator, by means of which all sub types of the structure can be detected.

Input Parameters:

Output Parameters:

| | |
|--|------------------------------|
| GDIDataTypeliterator*& pGDIDataTypeliterator | Reference to a type iterator |
|--|------------------------------|

Return value:

Error message of the GDI coordinator

3.25.2 GDIDATAGETSUBVALUEBYNAME

Prototype:

```
short GDIDATAGetSubvalueByName ( const char* Section,  
                                GDIDataBaseType*& pDataObj)
```

Task:

Returns the reference to a structure element. The basic type is returned as reference. The name is used as selector. The actual type of the structure element (if unknown) may be detected by means of *GDIDATAGetType*.

Input Parameters:

| | |
|---------------|---|
| char* Section | Section name of the referenced structure element. |
|---------------|---|

Output Parameters:

| | |
|----------------------------|----------------------|
| GDIDataBaseType*& pDataObj | Reference to a datum |
|----------------------------|----------------------|

Return value:

Error message of the GDI coordinator

3.25.3 GDIDATAGETSUBVALUEBYORDVALUE

Prototype:

```
short GDIDATAGetSubvalueByOrdValue (    unsigned long Section,  
                                         GDIDataBaseType*& pDataObj)
```

Task:

Returns the reference to a structure element. The basic type is returned as reference. A numeric value, which describes the position of the structure element within the structure, is used as selector. The actual type of the structure element (if unknown) may be detected by means of *GDIDATAGetType*.

Input Parameters:

| | |
|-----------------------|---|
| unsigned long Section | Ordinal value of the structure element. |
|-----------------------|---|

Output Parameters:

| | |
|----------------------------|----------------------|
| GDIDataBaseType*& pDataObj | Reference to a datum |
|----------------------------|----------------------|

Return value:

Error message of the GDI coordinator

3.26 INTERFACE GDIDATAUNION

Basis class

GDIDataBaseType

3.26.1 GDIDATAGETDATATYPEITERATOR

Prototype:

```
short GDIDATAGetDataTypeliterator ( GDIDataTypeliterator*&  
                                     pGDIDataTypeliterator)
```

Task:

Generates a type iterator, by means of which all sub types of the structure can be detected.

Input Parameters:

Output Parameters:

| | |
|--|------------------------------|
| GDIDataTypeliterator*& pGDIDataTypeliterator | Reference to a type iterator |
|--|------------------------------|

Return value:

Error message of the GDI coordinator

3.26.2 GDIDATAGETSWITCHTYPE

Prototype:

short GDIDataGetSwitchType (GDIDataBaseType*& pDataObj)

Task:

Generates a reference to pDataObj. The actual type of the switch value is detected by the service gdiGetType() in relation to the returned reference.

Input Parameters:**Output Parameters:**

| | |
|----------------------------|--|
| GDIDataBaseType*& pDataObj | reference of the data object, which describes the type of the Switch value |
|----------------------------|--|

Return value:

Error message of the GDI coordinator

3.26.3 GDIDATAGETSUBVALUEBYNAME

Prototype:

```
short GDIDATAGetSubvalueByName ( const char* Section,  
                                GDIDataBaseType*& pDataObj)
```

Task:

Returns the reference to a structure element. The basic type is returned as reference. The name is used as selector. The actual type of the structure element (if unknown) may be detected by means of *GDIDATAGetType*.

Input Parameters:

| | |
|---------------|---|
| char* Section | Section name of the referenced structure element. |
|---------------|---|

Output Parameters:

| | |
|----------------------------|----------------------|
| GDIDataBaseType*& pDataObj | Reference to a datum |
|----------------------------|----------------------|

Return value:

Error message of the GDI coordinator

3.26.4 GDIDATAGETSUBVALUEBYORDVALUE

Prototype:

```
short GDIDATAGetSubvalueByOrdValue (    unsigned long Section,  
                                         GDIDataBaseType*& pDataObj)
```

Task:

Returns the reference to a structure element. The basic type is returned as reference. A numeric value, which describes the position of the structure element within the structure, is used as selector. The actual type of the structure element (if unknown) may be detected by means of *GDIDATAGetType*.

Input Parameters:

| | |
|-----------------------|---|
| unsigned long Section | Ordinal value of the structure element. |
|-----------------------|---|

Output Parameters:

| | |
|----------------------------|----------------------|
| GDIDataBaseType*& pDataObj | Reference to a datum |
|----------------------------|----------------------|

Return value:

Error message of the GDI coordinator

3.27 INTERFACE GDIDATASEQUENCE

Basis class

GDIDataBaseType

3.27.1 GDIDATAGETSUBTYPE

Prototype:

```
short GDIDATAGetSubType (      unsigned long Index,  
                             GDIDataBaseType*& pDataObj)
```

The function is still contained in the API for reasons of the compatibility.
For Task and parameter see *GDIDATAGetCopySubTypeByIndex*.

3.27.2 GDIDATAGETCOPYSUBTYPEBYINDEX

Prototype:

```
short GDIDATAGetCopySubTypeByIndex ( unsigned long Index,  
                                      GDIDataBaseType*& pDataObj)
```

Task:

Generates a reference to RefDataObj. The actual type of the Array elements is detected by the service *GDIDATAGetType* in relation to the returned reference. A sub value object will be created by this activity as a copy.

Input Parameters:

| | |
|---------------------|---|
| unsigned long Index | Number of the element to be referenced within the array. The value has to be less than the size of the array. 0 references the first array element. |
|---------------------|---|

Output Parameters:

| | |
|----------------------------|--|
| GDIDataBaseType*& pDataObj | Reference to the type of the single elements, upon which the Array is based. |
|----------------------------|--|

Return value:

Error message of the GDI coordinator

3.27.3 GDIDATAGETSUBTYPEBYINDEX

Prototype:

```
short GDIDATAGetSubTypeByIndex (    unsigned long Index,  
                                   GDIDataBaseType*& pDataObj)
```

Task:

Generates a reference to RefDataObj. The actual type of the Array elements is detected by the service *GDIDATAGetType* in relation to the returned reference. The reference is valid up to a renewed call of this function.

Input Parameters:

| | |
|---------------------|---|
| unsigned long Index | Number of the element to be referenced within the array. The value has to be less than the size of the array. 0 references the first array element. |
|---------------------|---|

Output Parameters:

| | |
|----------------------------|--|
| GDIDataBaseType*& pDataObj | Reference to the type of the single elements, upon which the Array is based. |
|----------------------------|--|

Return value:

Error message of the GDI coordinator

3.27.4 GDIDATAGetSEQUENCETYPE

Prototype:

```
short GDIDATAGetSequenceType (    GDISEQUENCETYPE* pSeqType)
```

Task:

Returns the Type of the sequence.

Input Parameters:**Output Parameters:**

| | |
|---------------------------|----------------------|
| GDISEQUENCETYPE* pSeqType | Type of the sequence |
|---------------------------|----------------------|

Return value:

3.27.5 GDIDATAGETMAXELEMENTNUMBER

Prototype:

short GDIDATAGetMaxElementNumber (unsigned long* pMaxNumber)

Task:

Is used for the detection of the number of elements of an Array.

Input Parameters:**Output Parameters:**

unsigned long* pMaxNumber Number of elements of an Array

Return value:

Error message of the GDI coordinator

3.27.6 GDIDATAGETCURRENTSIZE

Prototype:

short GDIDATAGetCurrentSize (unsigned long* pCurrentSize)

Task:

Returns the number of valid data elements within a Sequence array. Starting from index 0, the data are located contiguously within the array.

Input Parameters:**Output Parameters:**

unsigned long* pCurrentSize Number of available array elements.

Return value:

Error message of the GDI coordinator

3.27.7 GDIDATASETCURRENTSIZE

Prototype:

short GDIDATASETCurrentSize (unsigned long CurrentSize)

Task:

Sets the number of valid data elements within a Sequence array Starting from index 0, the data are located contiguously within the array. If the number of data is increased, the contents of the added data are undefined.

Input Parameters:

unsigned long CurrentSize New number of valid data of a Sequence.

Output Parameters:**Return value:**

Error message of the GDI coordinator

3.27.8 GDIDATAGETELEMENTTYPE

Prototype:

GDIDATAGetElementType(GDIDATATYPE* pDataType)

Task:

Gets the data type of the sequence element (sub type object).

Input Parameters:**Output Parameters:**

GDIDATATYPE* pDataType Data type of the sequence sub type object

Return value:

Error message of the GDI coordinator

3.28 INTERFACE GDIDATAARRAY

Basis class

GDIDataBaseType

3.28.1 GDIDATAGETSUBTYPE

Prototype:

```
short GDIDATAGetSubType (      unsigned long Index,  
                             GDIDataBaseType*& pDataObj)
```

The function is still contained in the API for reasons of the compatibility.
For Task and parameter see *GDIDATAGetCopySubTypeByIndex*.

3.28.2 GDIDATAGETCOPYSUBTYPEBYINDEX

Prototype:

```
short GDIDATAGetCopySubTypeByIndex ( unsigned long Index,  
                                       GDIDataBaseType*& pDataObj)
```

Task:

Generates a reference to RefDataObj. The actual type of the Array elements is detected by the service *GDIDATAGetType* in relation to the returned reference. A sub value object will be created by this activity as a copy.

Input Parameters:

| | |
|---------------------|---|
| unsigned long Index | Number of the element to be referenced within the array. The value has to be less than the size of the array. 0 references the first array element. |
|---------------------|---|

Output Parameters:

| | |
|----------------------------|--|
| GDIDataBaseType*& pDataObj | Reference to the type of the single elements, upon which the Array is based. |
|----------------------------|--|

Return value:

Error message of the GDI coordinator

3.28.3 GDIDATAGETSUBTYPEBYINDEX

Prototype:

short GDIDATAGetSubTypeByIndex (unsigned long Index,
GDIDataBaseType*& pDataObj)

Task:

Generates a reference to RefDataObj. The actual type of the Array elements is detected by the service *GDIDATAGetType* in relation to the returned reference. The reference is valid up to a renewed call of this function.

Input Parameters:

| | |
|---------------------|---|
| unsigned long Index | Number of the element to be referenced within the array. The value has to be less than the size of the array. 0 references the first array element. |
|---------------------|---|

Output Parameters:

| | |
|----------------------------|--|
| GDIDataBaseType*& pDataObj | Reference to the type of the single elements, upon which the Array is based. |
|----------------------------|--|

Return value:

Error message of the GDI coordinator

3.28.4 GDIDATAGETMAXELEMENTNUMBER

Prototype:

unsigned long GDIDATAGetMaxElementNumber ()

Task:

Is used for the detection of the number of elements of an Array.

Input Parameters:**Output Parameters:****Return value:**

Number of elements of an Array.

3.28.5 GDIDATAGETELEMENTTYPE

Prototype:

GDIDATAGetElementType(GDIDATATYPE* pDataType)

Task:

Gets the data type of the array element (sub type object).

Input Parameters:**Output Parameters:**

GDIDATATYPE* pDataType Data type of the array sub type object

Return value:

Error message of the GDI coordinator

3.29 INTERFACE GDIIITERATOR

3.29.1 GDICOGETFIRSTITEM

Prototype:

short GDICOGetFirstItem (DCDBaselItem*& pInstance)

Task:

Returns the first instance of GDI objects and its derivatives, which are defined by the DCD and created by the coordinator.

Input Parameters:**Output Parameters:**

DCDBaselItem*& pInstance

reference of the instance

Return value:

Error message of the GDI coordinator

3.29.2 GDICOGETNEXTITEM

Prototype:

short GDICOGetNextItem (DCDBaselItem*& pInstance)

Task:

Returns the next instance of GDI objects and its derivatives, which are defined by the DCD and created by the coordinator.

Input Parameters:**Output Parameters:**

DCDBaselItem*& pInstance reference of the instance

Return value:

Error message of the GDI coordinator

3.31 INTERFACE GDIDeviceConnectionIterator

Basis class

GDIIterator

Task

The interface supports the delivery of instances of GDI Device Connections, which are created by the coordinator.

Output Parameters:

GDIDeviceConnection*& pGDIDeviceConnection reference of the instance

3.33 INTERFACE GDIDEVFUNCITERATOR

Basis class

GDIIterator

Task

The interface supports the delivery of device function instances, which are created by the coordinator.

Output Parameters:

GDIDeviceFunction*& pGDIDeviceFuncction reference of the instance

3.34 INTERFACE DCDDDEVFUNCITERATOR

Basis class

GDIIterator

Task

The interface supports the delivery of instances, which are created by the coordinator and administers defined DCD device functions.

Output Parameters:

| | |
|---|---------------------------|
| DCDDDeviceFunction*& pDCDDDeviceFuncction | reference of the instance |
|---|---------------------------|

3.35 INTERFACE DCDCOMOBJECTITERATOR

Basis class

GDIiterator

Task

The interface supports the delivery of instances, which are created by the coordinator and administers defined DCD communication objects.

Output Parameters:

DCDComObject*& pDCDComObject reference of the instance

3.37 INTERFACE DCDOOPERATIONITERATOR

Basis class

GDIiterator

Task

The interface supports the delivery of instances, which are created by the coordinator and administers defined DCD operations.

Output Parameters:

| | |
|------------------------------|---------------------------|
| DCDOperation*& pDCDOperation | reference of the instance |
|------------------------------|---------------------------|

3.38 INTERFACE GDIOPERATIONITERATOR

Basis class

GDIIterator

Task

The interface supports the delivery of operation instances, which are created by the coordinator.

Output Parameters:

| | |
|------------------------------|---------------------------|
| GDIOperation*& pGDIOperation | reference of the instance |
|------------------------------|---------------------------|

3.39 INTERFACE GDIDATATYPEITERATOR

Basis class

GDIIterator

Task

The interface supports the delivery of data type instances, which are created by the coordinator and supports the transfer of data between application and coordinator.

Output Parameters:

GDIDataBaseType*& pGDIDataBaseType reference of the instance

4 Definitions of Datatype

4.1 COMOBJECTTYPE

The type COMOBJECTTYPE specifies the type of the items of the respective device function.

There are two types. These are parameter(gdicot_Parameter) and attribute(gdicot_Attribute).The data type is defined :

```
enum COMOBJECTTYPE
{
    gdicot_Attribute,    //the member of the device function is an attribute
    gdicot_Parameter    //the member of the device function is an parameter
};
```

| name | description |
|------------------|---|
| gdicot_Attribute | the member of the device function is an attribute |
| gdicot_Parameter | the member of the device function is an parameter |

4.2 GDIACCRIGHTS

The type GDIACCRIGHTS defines the rights of access on a communication object.
The data type is defined :

```
enum GDIACCRIGHTS
{
    gdiarReadWrite,
    gdiarReadOnly    //communicationobject is read only
};
```

| name | description |
|----------------|---|
| gdiarReadWrite | access to communication object: GDI_Read/GDI_Write |
| gdiarReadOnly | access to communication object: only GDI_Read |

4.3 GDIERROR

The type GDIERROR contains the error messages of the executed GDI-API function [4] and supplies (if available) the appropriate DIT string back.

The data type is defined:

```
struct GDIERROR
{
    short Returncode;           //returncode of the call of device driver function (if called)
    unsigned short qual;       //only assigned if Returncode = COM_ERR otherwise 0
    unsigned short grade;      //only assigned if Returncode = COM_ERR otherwise 0
    unsigned short code;       //only assigned if Returncode = COM_ERR otherwise 0
    char ErrorString[256];     //forms from DIT (if available, otherwise empty string)
};
```

| type | name | description |
|----------------|-------------|--|
| short | Returncode | returncode of the GDI driver |
| unsigned short | qual | errorcode of qual transmitted by the gdi driver |
| unsigned short | grade | errorcode of grade transmitted by the gdi driver |
| unsigned short | code | errorcode of code transmitted by the gdi driver |
| char[256] | ErrorString | description of the error (DIT file) |

4.4 GDIDITERERRORPARAMETER

This type stores the errorcodes of a gdi device driver and a replace or errortext. The replace text is set by an application.

```
struct GDIDITERERRORPARAMETER
{
    unsigned short qual;
    unsigned short grade;
    unsigned short code;
    char ErrorString[256];
};
```

| type | name | description |
|----------------|-------------|--|
| unsigned short | qual | errorcode of qual transmitted by the gdi driver |
| unsigned short | grade | errorcode of grade transmitted by the gdi driver |
| unsigned short | code | errorcode of code transmitted by the gdi driver |
| char[256] | ErrorString | description of the error (DIT file) |

4.5 ACCEPTECCURED

Returns the information, if an Accept has been carried out since the last test or `gdiWriteValue`.

```
enum ACCEPTECCURED
{
    gdiac_noAccept,
    gdiac_Accept
};
```

| name | description |
|-----------------------------|--------------------|
| <code>gdiac_noAccept</code> | no Accept occurred |
| <code>gdiac_Accept</code> | Accept occurred |

4.6 INFREPORTOCCURED

Returns the information, if an Information Report has been carried out since the last test or `gdiReadValue`.

```
enum INFREPORTOCCURED
{
    gdiir_noInfReport,
    gdiir_InfReport
};
```

| name | description |
|--------------------------------|-------------------------------|
| <code>gdiir_noInfReport</code> | no InformationReport occurred |
| <code>gdiir_InfReport</code> | InformationReport occurred |

4.7 GDIDATATYPE

GDIDATATYPE specifies all occurring DCD data type definitions.
The data type is defined :

```
enum GDIDATATYPE
{
    NoDefinition =0,
    gdidt_Char =1,
    gdidt_Boolean =2,
    gdidt_Short =3,
    gdidt_UShort =4,
    gdidt_Enum =5,
    gdidt_Long =6,
    gdidt_ULong =7,
    gdidt_Float =8,
    gdidt_Double =9,
    gdidt_Array =20,
    gdidt_Sequence =21,
    gdidt_Struct =23,
    gdidt_Union =24,
    gdidt_Octet =25,
    gdidt_DevFuncRef=30
};
```

4.8 DATAINSTANCEVERSION

Returns the description of the derivation of the datum.

```
enum DATAINSTANCEVERSION
{
    div_InitData,
    div_CreateParameter,
    div_Parameter,
    div_Attribute,
    div_InValue,
    div_OutValue
};
```

| name | description |
|---------------------|--------------------|
| div_InitData | InitData (VD) |
| div_CreateParameter | CreateParameter |
| div_Parameter | Parameter |
| div_Attribute | Attribute |
| div_InValue | InValue |
| div_OutValue | Out Value |

4.9 GDISEQUENCETYPE

Returns the type of the sequence.

```
enum GDISEQUENCETYPE  
{  
    st_Limited,  
    st_Unlimited  
};
```

| name | description |
|--------------|--------------------|
| st_Limited | limited sequence |
| st_Unlimited | unlimited sequence |

4.10 GDIENUMERATIONMEMBER

Returns the a element of an enumeration.
The data type is defined:

```
struct GDIENUMERATIONMEMBER
{
    short Value
    char Identifier[50];
};
```

| type | name | description |
|-------------|-------------|---|
| short | Value | value of the member of the enumeration |
| char[50] | Identifier | identifier of the member of the enumeration |

4.11 GDICOMMTYPE

The type GDICOMMTYPE specifies the type of communication between device and GDI driver.

Possible communication types are synchronous and asynchronous.

The data type is defined:

```
enum GDICOMMTYPE
{
    gdictSYNC=0,           //requests a synchronous communication
    gdictASYNC=1          //requests a asynchronous communication
};
```

| name | description |
|-------------|---|
| gdictSYNC | requests a synchronous communication between GDI driver and device |
| gdictASYNC | requests a asynchronous communication between GDI driver and device |

4.12 GDIIDENT

Returns specific details to a VD.
The data type is defined:

```
struct GDIIDENT {  
    long DeviceVersion;  
    char DriverName[256];  
    long DriverVersion;  
    char FactoryName[256];  
};
```

4.13 GDISTAT

Returns the state of the VD.
The data type is defined:

```
struct GDISTAT
{
short log;
short phys;
short phase
GDIERROR detail;
};
```

4.14 INSTANCEITEM

The type INSTANCEITEM specifies the name of the created GDI Driver instance. The instance is created by the coordinator.

```
typedef char INSTANCEITEM [512];
```

4.15 IDENTITEM

The type IDENTITEM specifies the name of the identifier of a GDI item. The GDI items are defined by the dcd file. GDI Items are modules, interfaces, attributes, parameter and operations.

```
typedef char IDENTITEM [256];
```

5 Error codes

| Var | Code | Destignation |
|--------------------------------|-------|--|
| CO_OK | 0 | No error |
| COERR_SYNC_OK | 0 | No error with synchronous communication |
| COERR_ASYNC_OK | 1 | No error with asynchronous communication |
| COERR_DCD | -209 | Can not load DCD file |
| COERR_LICENSE | -999 | No license for using the Coordinator |
| COERR_LOADLL | -1000 | Driver could not be loaded |
| COERR_DLLFUNC | -1001 | Function could not be found in DLL |
| COERR_FREEDLL | -1002 | Driver could not be unloaded |
| COERR_MEMORY | -1003 | No memory could be reserved |
| COERR_TIMEOUT | -1004 | Timeout |
| COERR_FALSECTRLVD | -1005 | Control VD not found |
| COERR_VDCONNECTIONNOTFOUND | -1007 | VD Connection not found |
| COERR_GDIDRIVERERRORMSG | -1008 | An error message of the device driver is present. |
| COERR_CREATEDEVFUNC | -1009 | Device function could not be produced |
| COERR_DELETEDEVFUNC | -1010 | Device function could not be deleted |
| COERR_INITIALIZEDEVFUNCMEMBER | -1011 | Object of the device function could not be initialized |
| COERR_UNKNOWTRF | -1012 | Unknown transition function |
| COERR_EXECUTETRANSITION | -1013 | Error during the execution of the transition function |
| COERR_CREATEPARAMNOTFOUND | -1014 | Create parameter was not found |
| COERR_FALSESEVFUNC_CREATEPARAM | -1015 | False create parameter of the device function |

| Var | Code | Destignation |
|-------------------------------|-------------|--|
| COERR_DEVFUNCNOTFOUND | -1016 | device function not found |
| COERR_COMOBJNOTFOUND | -1017 | device function member not found |
| COERR_NOINFORMATIONREPORTFUNC | -1018 | InformationReport function not defined |
| COERR_EXCEPTIONBYGDI | -1019 | GDI driver releases a exception |
| COERR_TYPEMISMATCH | -1020 | Type mismatch |
| COERR_INVALIDOBJECT | -1021 | Invalid object |
| COERR_NOTAVAILABLE | -1022 | object or functionality not available |
| COERR_DRIVERNOTFOUND | -1023 | GDI driver not found |
| COERR_ATTACH | -1024 | Errormessage by the GDI driver during the function call GDI_Attach |
| COERR_NODATAINFO | -1025 | No information about a data object |
| COERR_DATAINDEX | -1026 | Invalid index |
| COERR_MODULNOTFOUND | -1027 | Module not found |
| COERR_FILENOTFOUND | -1028 | File not found |
| COERR_MACROPARSER | -1029 | Errormessage of the macro parser |
| COERR_SYNTAX | -1030 | Syntax error |
| COERR_DATAOBJNOTFOUND | -1031 | Data object not found |
| COERR_CREATEFILE | -1032 | Coordinator can not create a macrofile or a logfile |
| COERR_OBJECTLOCKED | -1033 | GDI object is locked. The coordinator can not run again |
| COERR_UNKNOW | -1999 | Unknown error |
| COERR_ACCESS_VIOLATION | -2000 | Access fault in the GDI driver |
| COERR_FLT_DIVIDE_BY_ZERO | -2006 | Float divide by zero |

| | | |
|-----------------------------|-------------|---|
| COERR_FLT_INVALID_OPERATION | -2008 | Invalid operation |
| Var | Code | Destignation |
| COERR_INT_DIVIDE_BY_ZERO | -2012 | Integer divide by zero |
| COERR_INT_OVERFLOW | -2013 | Integer overflow |
| COERR_REQUESTINDEX | -3002 | The Iterator can not found the first or the next object. Are no more objects available. |
| | | |
| DITMERR_OPEN | -5005 | Server/Coordinator can not open DIT file |

Books

Coordinator Services Release, Revision 1, ASAM-GDI Version 4.20 - 01.08.2002

Specification of the Device Capability Description of an ASAM GDI-Driver, ASAM-GDI Version 4.20 – 09.02.2001

Specification of the Interface of an ASAM GDI-Driver, ASAM-GDI Version 4.20 - 09.02.2001

Generic Device Interface Version 4.3.2 Specification – 15.02.2005

ASAM GDI Version 4.4 Specification – 31.01.2008