

# ASAM GDI – Standard Extension

## API Documentation



rd electronic GmbH  
Zweigstelle Dresden  
Bernhardstraße 70  
01187 Dresden

Tel. +49 351-6563-56-4

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

---

Document Number:	ASAM-GDI-StdExtension-UserMan
Issue:	2.0.0.0
Status:	draft
Last Modification:	2017-06-28
Created on:	2006-07-12
Author:	SBa

---

## **Important Notice**

THIS DOCUMENT CONTAINS INFORMATION PROTECTED BY COPYRIGHT LAW. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY ANY MEANS, ELECTRONIC, MECHANICAL, PHOTOCOPYING, RECORDING, OR OTHERWISE, WITHOUT THE PRIOR WRITTEN PERMISSION OF THE PUBLISHER.

THE AUTHOR RESERVES THE RIGHT TO CHANGE THE INFORMATION BEING PART OF THIS DOCUMENT WITHOUT ANY NOTICE.

THE INFORMATION CONVEYED IN THIS DOCUMENT HAS BEEN CAREFULLY REVIEWED AND BELIEVED TO BE ACCURATE AND RELIABLE; HOWEVER, NO RESPONSIBILITY IS ASSUMED FOR INACCURACIES IN THIS DOCUMENT. THE AUTHOR IS NOT RESPONSIBLE FOR ERRORS IN THIS DOCUMENT OR FOR ANY DAMAGE CAUSED AT RANDOM OR FOR CONSEQUENTIAL LOSS IN CONNECTION WITH THE APPLICATION OF THIS DOCUMENT.

---

## Document Change History

Version	Date	Changed by	Modification
1.0.0.0	2006-07-13	BA	<ul style="list-style-type: none"><li>• Initial version</li></ul>
1.0.0.1	2006-09-22	BA	<ul style="list-style-type: none"><li>• All parts are described in the specification are skipped</li></ul>
1.0.1.0	2010-09-27	SBa	<ul style="list-style-type: none"><li>• Konfiguration file name adapted</li></ul>
2.0.0.0	2017-06-28	SBa	<ul style="list-style-type: none"><li>• Change to API Documentation</li><li>• Extraction of configurations in separated document</li></ul>

---

## Contents

1	General	6
1.1	Identification	6
1.2	Purpose	6
1.3	Referenced Documents	6
2	Resource Objects	7
2.1	General Protection of Resource Objects	7
2.2	Synchronization of Functions	7
3	Interface Types COM/ECOM	8
3.1	ext_initiate	8
3.1.1	User Messages	8
3.1.2	GDI Symbol Mapping	8
3.2	ext_open	8
4	Interface Type IP4	9
4.1	ext_initiate	9
5	Interface Type EIP4	10
5.1	ext_initiate	10
6	Interface Type ECOM	11

---

## Figures and Tables

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

---

## 1 General

### 1.1 Identification

This document describes behavior of the Standard Extension that are not prescribed within the standard document [1].

### 1.2 Purpose

It is important for the application programmer to know different states of Standard Extension resources, general rules for using. These parts are not standardized by the ASAM-GDI specification but documented here.

### 1.3 Referenced Documents

- [1] ASAM GDI Specification „ASAM\_GDI\_4\_3\_2\_Specification.pdf“

---

## 2 Resource Objects

This chapter lists resource objects of the Standard Extension. Resource objects represent Interface Types and Connections. Construction and destruction of Interface Types are supported via the API functions `ext_initiate` and `ext_conclude`. Construction and destruction of Connections are supported via the API functions `ext_open` and `ext_close`. These objects are identified by the handles are returns or parameters of these functions.

### 2.1 General Protection of Resource Objects

All resource objects support multi thread access. If a resource is released by a call of `ext_conclude` or `ext_close` but currently used by another thread then it isn't destroyed directly. The thread that requires the destruction of a resource object sets a request, deregisters the resource by the Standard Extension and returns immediately. The resource is not available any longer. The last using thread destroys the resource object.

This implementation avoids access violations by early releasing of resources.

### 2.2 Synchronization of Functions

The Standard Extension synchronizes the API functions that access the same resource object. It is possible to use a resource object simultaneously by using functions that don't change properties of a resource object. The functions `ext_open` and `ext_config` are not callable simultaneously on a resource object. Other threads that access to this resource object during execution of these functions are blocked.

## 3 Interface Types COM/ECOM

This chapter lists API functions of the Standard Extension that interprets an external configuration, an external parameterisation or have a special behaviour.

### 3.1 `ext_initiate`

By a call of this function the handling of user messages and symbol mapping for the Interface Types COM and ECOM are parameterised.

#### 3.1.1 *User Messages*

When a COM/ECOM Interface Type is initialized via `ext_initiate` then the environment variable `GDI_PAEXT_COM_MSG_ON_ERROR` is interpreted.

The environment variables switch the user messages on any errors on the specific Interface Type. If the variable is settled to `true` and an error happens then a message box comes up.

If the variable is not settled or empty then the current setting is unchanged. The default value is `false`.

Note that the Platformadapter calls the API function `ext_initiate`. Perhaps the function will be called only at the first initialization of the Interface Type and not by following referencing of this Interface Type.

#### 3.1.2 *GDI Symbol Mapping*

When a COM/ECOM Interface Type is initialized via `ext_initiate` then the environment variable `GDI_PAEXT_DIR` is interpreted and the symbol-mapping table initialized by interpretation of the configuration file.

When the function `ext_open` (3.2) is called the symbol mapping is used to find the special file for communication. The path and name of the special file depends on the operating system.

### 3.2 `ext_open`

When a periphery connection is opened by a call of `ext_open` then the name (`pszName`) is a member of the structure `IO_CONFDAT` is mapped to the special file name on the operating system. The symbol mapping is loaded by a call of `ext_initiate` (3.1.2) before.



## 4 Interface Type IP4

This chapter lists API functions of the Standard Extension that interprets an external configuration, an external parameterisation or have a special behaviour.

### 4.1 `ext_initiate`

When an IP4 Interface Type is initialized via `ext_initiate` then the environment variable `GDI_PAEXT_IP_MSG_ON_ERROR` is interpreted.

The environment variable switches the user messages on any errors on the Interface Types `IP4/EIP4`. If the variable is settled to `true` and an error happens then a message box comes up.

If the variable is not settled or empty then the current setting is unchanged. The default value is `false`.

Note that the Platformadapter calls the API function `ext_initiate`. Perhaps the function will be called only at the first initialization of the Interface Type and not by following referencing of this Interface Type.

## 5 Interface Type EIP4

This chapter lists API functions of the Standard Extension that interprets an external configuration, an external parameterisation or have a special behaviour.

### 5.1 `ext_initiate`

When an EIP4 Interface Type is initialized via `ext_initiate` then the environment variable `GDI_PAEXT_IP_MSG_ON_ERROR` is interpreted.

The environment variable switches the user messages on any errors on IP4/EIP4 Interface Types. If the variable is settled to `true` and an error happens then a message box comes up.

If the variable is not settled or empty then the current setting is unchanged. The default value is `false`.

Note that the Platformadapter calls the API function `ext_initiate`. Perhaps the function will be called only at the first initialization of the Interface Type and not by following referencing of this Interface Type.

---

## 6 Interface Type ECOM

The Standard Extension supports this Interface Type only on Windows systems.

A call of `io_initiate/ext_initiate` on Linux to this Interface Types returns the error code `IOE_TYPE_ID`.