INCA ASAM MCD-3MC V1.0.1 Interface

User Guide

Copyright

Contents

1 Safety and Privacy Information

1.1 Intended Use

INCA and INCA add-ons are developed and approved for automotive applications and procedures as described in the user documentation for INCA and INCA add-ons.

INCA and the INCA add-ons are intended to be used in industrial labs and in test vehicles.

ETAS GmbH cannot be made liable for damage that is caused by incorrect use and not adhering to the safety information.

1.2 Target Group

This software product and this user guide address qualified personnel working in the fields of automotive ECU development and calibration, as well as system administrators and users with administrator privileges who install, maintain, or uninstall software. Specialized knowledge in the areas of measurement and ECU technology is required.

1.3 Classification of Safety Messages

The safety messages used here warn of dangers that can lead to personal injury or damage to property:

> ⚠️ **DANGER**
>
> DANGER indicates a hazardous situation that, if not avoided, will result in death or serious injury.

> ⚠️ **WARNING**
>
> WARNING indicates a hazardous situation that, if not avoided, could result in death or serious injury.

> ⚠️ **CAUTION**
>
> CAUTION indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.

> **NOTICE**
>
> NOTICE indicates a situation that, if not avoided, could result in damage to property.

1.4 Safety Information

> ⚠ **WARNING**
>
> **Risk of unexpected vehicle behavior**
>
> Calibration activities influence the behavior of the ECU and the systems that are connected to the ECU.
>
> This can lead to unexpected vehicle behavior, such as engine shutdown as well as breaking, accelerating, or swerving of the vehicle.
>
> Only perform calibration activities if you are trained in using the product and can assess the possible reactions of the connected systems.

Adhere to the instructions in the ETAS Safety Advice and the safety information given in the online help and user guides.

Open the ETAS Safety Advice in the INCA help menu **? > Safety Advice**.

1.5 Privacy Notice

Note that personal data is processed when using INCA. As the controller, the purchaser undertakes to ensure the legal conformity of these processing activities in accordance with Art. 4 No. 7 of the General Data Protection Regulation (GDPR/EU). As the manufacturer, ETAS is not liable for any mishandling of this data.

For further information, refer to the INCA online help.

2 About INCA ASAM MCD-3MC V1.0.1

The ASAM standard provides several ASAM MCD-3MC interfaces. The following figure gives an overview of the ASAM MCD-3MC interfaces.
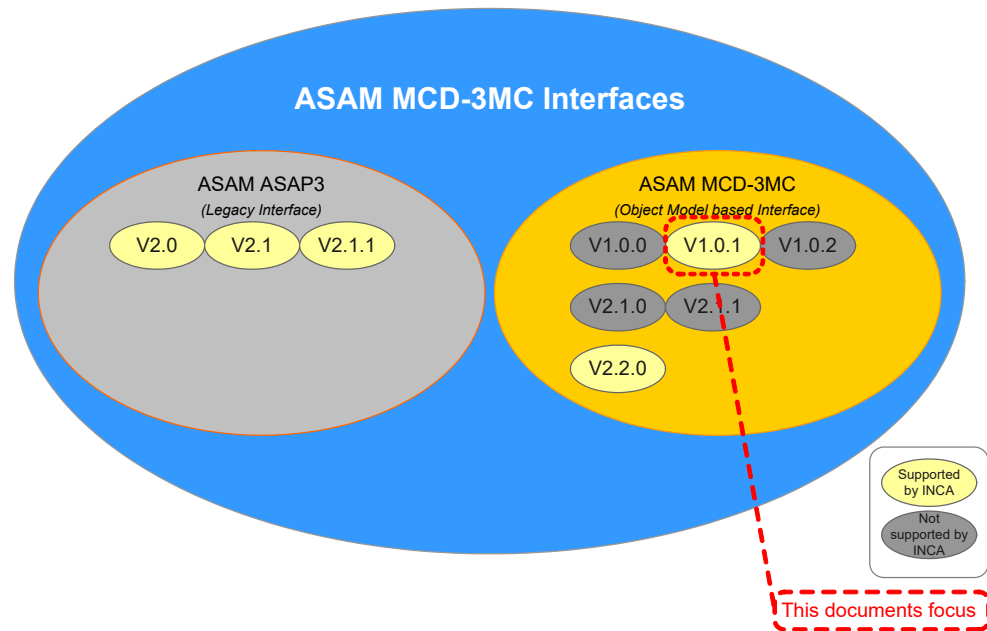
Fig. 1-1   Overview ASAM-MCD-3MC interfaces

This document describes the INCA specific implementation of the ASAM MCD-3MC server, called ASAM MCD 3MC 4 INCA. It is assumed that the reader is familiar with the ASAM MCD-3MC V1.0.1 Specification, available at www.asam.net or at ETAS. Only areas, where INCA specific implementation details are necessary to understand, are documented in this paper.

ASAM MCD-3MC defines different classes and names for entities that are also found in INCA, e.g., "Project" in INCA is not the same as a "Project" in ASAM MCD-3MC. In addition, some entities are modeled differently than in INCA and are new for an INCA user, e.g., the "Logical Link". The chapter "Definition of Terms Definition of Terms" gives some useful information for end users, that is, operators of control systems or calibration systems that are connected to INCA via ASAM MCD-3MC . Chapter "3 Implementation Specific Details" describes in detail the specific implementation of the different classes. Chapter "With Windows XP and higher operating system versions, the management console applet is displayed. Chose the Component services node and below this the computer node. There you will see the Workstation node. The properties dialog of the Workstation node provides tabs for "Default Properties", "Default Protocols" and "COM Security" (amongst others).
Below the Workstation node, there are four additional nodes. Here, only the "DCOM Configuration" node is important and can be used for the same purpose as the "Applications" tab in Windows 2000." gives additional information about the general behavior of ASAM and describes the additional "Settings" class.

2.1　Document Conventions

If there exists a collection for a specific object, the heading contains a plural "(s)" after the name, indicating this object is managed via a collection.

2.2　Definition of Terms

Project Management

The term "Project" in ASAM MCD-3MC has a different meaning than the same term has in INCA. An Inca Project is an A2L description file along with associated binaries, while a Project in ASAM MCD-3MC is a collection of all description files, binary files, configured devices and configuration information that is required for a certain setup.

Here is a table that describes the different names and meanings of INCA and ASAM MCD-3MC entities:

ASAM MCD-3MC Name	INCA Name	Description
MCProject	Workspace	The INCA workspace and the folder, in which the workspace resides, including all subfolders, build up the *MCProject*. The name of the *MCProject* is made up from the workspace name including the path within the database, e.g., "DEFAULT\Workspace".
MCDbLogicalLink	Description file and device in workspace	The *MCDbLogicalLink* is the combination of the description file, the device and the device type (or protocol type). The name of the *MCDbLogicalLink* is created from the names of its parts: ‹ description file name with path in INCA DB ›_‹ device name in workspace ›_‹ protocol type ›, e.g., "DEFAULT\0400_ETKC:1_ETK". See chapter "3.2.10 MCDbPhysicalVehicleLinkOrInterface(s)" for a detailed list of INCA devices and associated protocol types.
MCDbPhysicaVehicleLinkOrlInterface	Device	In MCD3, this is only used for informational purposes, e.g., to see the type of the device and the name.
MCDbBinary	Dataset	Contains either only data or data and code of the ECU.

Tab. 1-1　　Different names and meanings of INCA and ASAM MCD-3MC entities

INCA supports not only ECU-devices, e.g., those connected via ETK, CCP or XCP, but also pure measurement devices and some special devices. Pure

measurement devices are accessed exactly like ECU devices. Special devices are sometimes handled slightly different:

Device Type in INCA	Device Name in ASAM MCD-3MC	Description
RAMCal	The name of the standard device with suffix "#RAMCal"	Allows calibration of measurement labels. Not supported in MCD 3MC.
MeasureCal	The name of the standard device with suffix "#MeasureCal"	Allows measurement of calibration labels. Not supported in MCD 3MC.
CalcDev	"CalcDev"	Allows access to the calculated variables in the current INCA Experiment. LogicalLinks with this device are only available as long as the experiment, that is currently assigned to the selected MCDProject (= INCA workspace), contains definitions for calculated labels.
FULI/EtherCAT/MCE enabled device	The name of the standard device with suffix "#FULI"	Allows configuration of measurement and calibration labels that are to be sent via EtherCAT. In case a device, that supports EtherCAT, is connected to INCA, two logical links are available: one standard logical link and one with the suffix "#FULI" after the device name part, e.g., "DEFAULT\0400_ETKC:1#FULI_ETK". In case measurement or a calibration shall be done via EtherCAT, the logical links with the suffix #FULI must be used. All Variables added to such a logical link will be available via EtherCAT. If a device is accessed both via a standard logical link and the corresponding #FULI logical link, be sure to use the same binary for both, otherwise creation of the second logical link will fail. Writing calibration via one logical link will also change the calibration of the other logical link. If the option "Always use FULI device if available" is checked, all devices, that would normally appear with a second logigal link with "#FULI" appended, now are shown only once and without "#FULI" appended, but with the FULI functionality enabled. This is to allow for

Device Type in INCA	Device Name in ASAM MCD-3MC	Description
		easy migration to FULI/MCE systems. All rules for FULI logical links also apply to these logical links.

Tab. 1-2   Special devices

There are several options introduced that influence the behaviour of ASAM MCD 3MC 4 INCA. See chapter "4.2 Settings" and the online help for further information.

Calibration

In the current version of ASAM MCD-3MC scalars, curves and maps are supported. It is not possible to explicitly instantiate calibration objects of the type of VAL_BLK (vectors or matrices). This was shifted for a later version. Of course, the implicit usage of vectors and matrices together with curves or maps (X-Axis, Y-Axis, Curve/Map values itself) is possible.

The current ASAM MCD-3MC standard is based on ASAM MCD 2MC version 1.4 which doesn't contain Curve Axes. This implies that Curve Axes are not supported in our implementation.

According to the old ASAP3 server it's not possible to change rescale objects. The read command can be used without a restriction.

In case of verbal conversions (COMPU_VTAB, COMPU_VTAB_RANGE) strings will not be supported by the current ASAM MCD-3MC standard. Therefore, the ECU values will be used for read and write activities when the representation type is eRT_PHYSICAL (compatible to the old ASAP3 implementation).

With some Settings in the options dialog box the user can configure the behaviour of the ASAM MCD-3MC server respectively INCA.

With the setting "Show edited values in INCA" it can be selected if the characteristics which will be used for read or write operations shall be displayed or not.

When the second option "Use hard bound limitation" is set, INCA will be configured to accept values outside the weak bounds. In this case all values between the hard bounds will be accepted in INCA. If there is a value outside the hard bounds, it will be set to the hard bound automatically. INCA will be configured (with an INCA COM API call) to use the hard bounds as soon as at least one value is outside the weak bounds.

With the user option "Use 1Dimensional array of 1Dimensional arrays to return matrix values" the user can decide if the values of matrices (in case of reading maps) will be returned to the client in a two-dimensional array or in a one-dimensional array of one-dimensional arrays. For write activities this option will not be regarded because the format of the values which must be written can be derived directly out of the command.

Error messages will be displayed in the log window of the ASAM MCD-3MC server when the access of a client fails e.g., when a characteristic or one of its

components is write-protected (because of an active Reference Page in INCA) and the client tries to modify values.

For users who switch from the old ASAP3 server to the ASAM MCD-3MC server the following table provides an overview of how the functions of the old server are mapped to the new methods:

Designation ASAM MCD-3MC	Designation ASAP3
Scalars	
ReadVariant() Read() (this method is not implemented yet)	GET PARAMETER FROM AP-S
WriteVariant() Write() (this method is not implemented yet)	SET PARAMETER ON AP-S
Curves	
Read()	GET LOOKUP TABLE FROM AP-S GET LOOKUP TABLE VALUE
Write()	PUT LOOKUP TABLE TO AP-S INCREASE LOOKUP TABLE SET LOOKUP TABLE
Maps	
Read()	GET LOOKUP TABLE FROM AP-S GET LOOKUP TABLE VALUE
Write()	PUT LOOKUP TABLE TO AP-S INCREASE LOOKUP TABLE SET LOOKUP TABLE

Tab. 1-3   Overview of how the functions of the old server are mapped to the new methods

The characteristic access methods of the new standard cover the complete functionality of the old commands and even more because it's possible now to read/write ranges of X/Y-Axis or Curve/Map values. Additionally, it's possible now to access each component of a characteristic independent of the other components of that characteristic.

Measurement

The current INCA design imposes some limitations to the collectors within ASAM MCD 3MC 4 INCA:

It is possible to start and stop Collectors independently, but it is not possible to configure any Collector while another one is running (i.e., Check/Change is not allowed).

The maximum update rate for a Collector is 100ms (by default, tunable within INCA options). Measurement variables wired in a faster raster (e.g., 10ms) are transported in blocks, i.e., for a 10ms variable a block of 10 new values is available every 100ms. For clients polling with a rate < 100ms always that latest available value of the current block is delivered. Therefore, polling with rates < 100ms does not make sense (It only generates higher workload on server and client side).

If during measurement Buffer overflows occur very often, this is an indicator that the system workload is too high. In some cases, it might be possible to solve the problem by increasing the Buffer size, but most often it is necessary to reduce the number of wired values.

Possible reasons for overflows are:

– Server is not fast enough (faster PC required)
– Buffer size on Server is too small (increase Buffer size)
– Client is not fast enough (faster PC required / optimization of client-side code)
– Network overload (use a dedicated connection between Server and Client, use at least a 100 Mbit cable connection because WLAN has to high latency times)

Database

The current ASAM MCD 3MC 4 INCA version does not support all database items. Chapter "3.2.4, *Database objects*", on page 50 shows the details for each database object.

Miscellaneous

ASAM MCD 3MC 4 INCA provides several options. Since it is sometimes desirable to access these options also from the client, a new, proprietary settings class was introduced that provides the settings also to the client. See chapter "4.2 Settings" for details.

2.3 Further Information

Information on the ASAM MCD-3MC for INCA server is provided in the server GUI's online help.

3 Configuration of the ASAM MCD-3MC Communication

The communication between the test bench computer and INCA on the DCOM level of the operating system MS Windows is based on the ASAM MCD-3MC standard.

This section describes the operating system's configuration for the communication between the test bench as a client and INCA as the server. The description is divided into the following sections:

- Section 2.1 Requirements, describes general conditions under which you can connect the test bench to ASAM MCD-3MC.
- Section 2.2, DCOMCNFG Configuration Software describes the configuration software used for setting the DCOM parameters.
- Section 2.3 Configuration of the General DCOM Parameters, describes how the general communication parameters must be set.
- Section 2.4 IINCA Server Configuration, describes the computer configuration that runs the INCA software as a server application.
- Section 2.5 ASAM MCD-3MC Client Configuration, describes the computer configuration that runs your ASAM MCD-3MC client application.
- Section 2.6 Peer-To-Peer Connections, on page 35, describes special considerations that need to be observed with peer-to-peer connections.

3.1 Requirements

The test bench connection under ASAM MCD-3MC is recommended for computers running the Windows XP, Windows Vista or Windows 7 operating system.

The computers involved must be connected to each other either via a Windows NT server domain, an Active Directory server or within a peer-to-peer network. If there is a peer-to-peer connection between the computers, DCOM can be used only if the same user is logged in on both computers and the user also belongs to the "standard user" user group (or better) on both computers.

The INCA server requires the INCA software Release 5.0 or higher. The user option to use the new ASAM MCD-3MC standard must be enabled. The software requirements for the client are described in Section 2.5.1, Preparations.

To configure the DCOM parameters on the client and server, you need administrator rights on each computer. The configuration software for the DCOM parameters is installed under Windows XP, Windows Vista, and Windows 7 by default as part of the operating system. For detailed information on DCOM/COM, refer to the online help of your operating system.

To execute a DCOM connection, up to Windows XP it is OK to use an administrator account on the client and the server. Starting with Windows Vista, the new Microsoft security concept suggests running both sides under a non-Administrator account. While it is possible to also run both sides under an administrator account, that has User Authentication Control (UAC) turned off, this is not recommended at all.

Network Topology

It is generally suggested to provide a dedicated network connection for the ASAM MCD-3MC server and the client PC. This avoids interrupt from other traffic in the network:
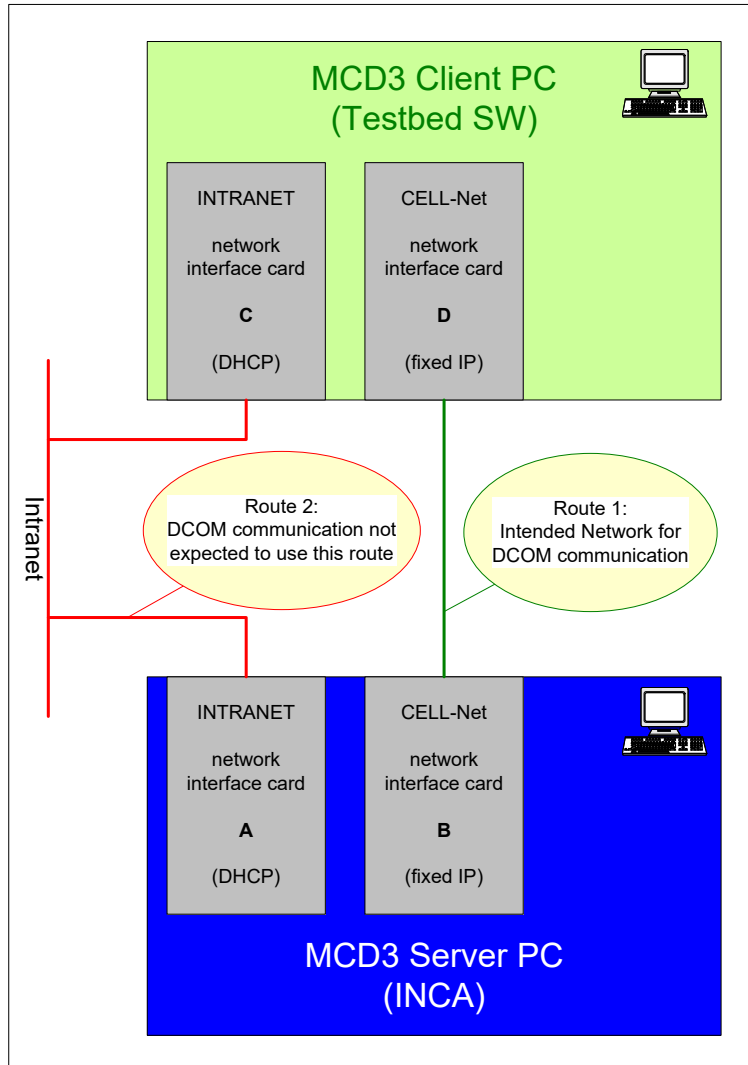
Fig. 2-1 Network technology

In case both PCs are in addition connected to a corporate LAN, problems arise because of the way DCOM resolves the addresses of both PCs: Even if the client specifies the connection to the server by a fixed IP address, both server and client will negotiate on a connection again based on a name lookup query. The order in which the connections are listed in this query is not known, so it is not possible to give precedence to the direct connection between client and server; it can easily happen that both will communicate over the corporate LAN instead.

A solution for this dilemma is to configure the IP addresses of both machines in their partner "hosts" file (or in the "lmhosts" file in case the LanManager protocol is enabled), because this entry has precedence over any other name server lookup.

The "hosts" file is located at [windows]\system32\drivers\etc .

On the INCA PC the host name of the MCD3 client PC and it's fixed IP address (network interface card D) must be entered.

On the MCD3 client PC the host name of the MCD3 server PC and it's fixed IP address (network interface card B) must be entered.

With this configuration, all DCOM communication is forced to use this defined route instead of any other possible one. Therefore, not MCD-related traffic between both PCs should be avoided.

This workaround was recommended also by Microsoft Support to overcome the described routing issue.

3.2 DCOMCNFG Configuration Software

The DCOMCNFG program is used for the configuration.

To Start the DCOM Configuration Software

1. Up to Windows XP: Choose the **Run** option in the **Start** menu.

   The "Run" dialog box is displayed.

   With Windows Vista and higher OS versions: Open the **Start** menu. The cursor is positioned in the search box.

2. Type in the **DCOMCNFG** command. In OS-Versions up to Windows XP: Click **OK**.

3. In Windows Vista and higher OS-Versions: Click the spyglass button or press Enter. Confirm the appearing UAC dialog with Continue.

   With Windows 2000, the "Properties of DCOM Configuration" dialog box is displayed. Configure all DCOM parameters in this dialog box.
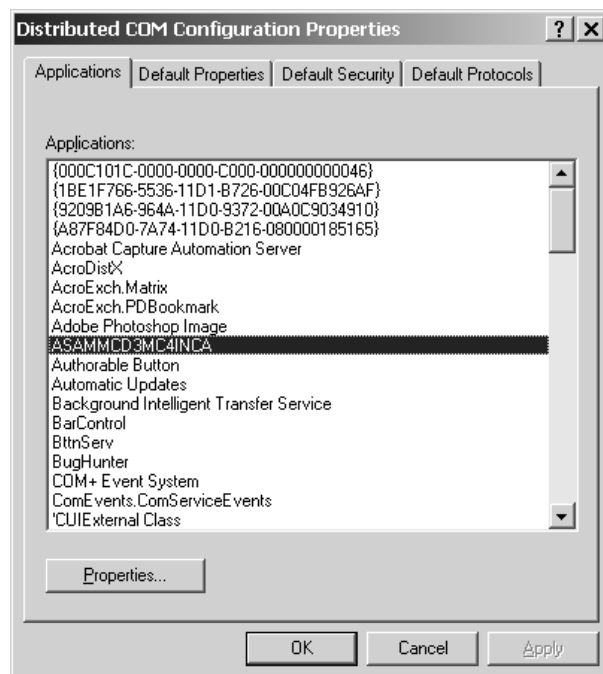
4. With Windows XP and higher operating system versions, the management console applet is displayed. Chose the Component services node and below this the computer node. There you will see the Workstation node. The properties dialog of the Workstation node provides tabs for "Default

Properties", "Default Protocols" and "COM Security" (amongst others). Below the Workstation node, there are four additional nodes. Here, only the "DCOM Configuration" node is important and can be used for the same purpose as the "Applications" tab in Windows 2000.

The "Properties of DCOM Configuration" dialog box contains four tabs on which the parameter setup functions are grouped:

- Use the "Default Properties," "Default COM Security," and "Default Protocols" tabs to set general parameters that apply to the computer.
- Use the "Applications" tab to specify application-specific parameters that are valid only for the selected application in its function as a client or server and may override your general setups.

Although the configuration is slightly different depending on the operating system being used, the DCOMCNFG configuration program is used under all versions of MS Windows.

3.3   Configuration of the General DCOM Parameters

Before beginning the various client and server configurations, you need to enable DCOM on all computers involved and set up general communication parameters.

3.3.1   Configuration of DCOM Parameters under Windows 2000

To Set Up General DCOM Parameters

1. Launch the DCOM configuration software as described above (Section 2.2, DCOMCNFG Configuration Software).
2. Select the **Default Properties tab.**
3. Enable the **Enable Distributed COM on this computer** option.
4. Set the value of the "Default Authentication Level" option to `Connect.`
5. Set the value for the "Default Impersonation Level" option to `Identify.`
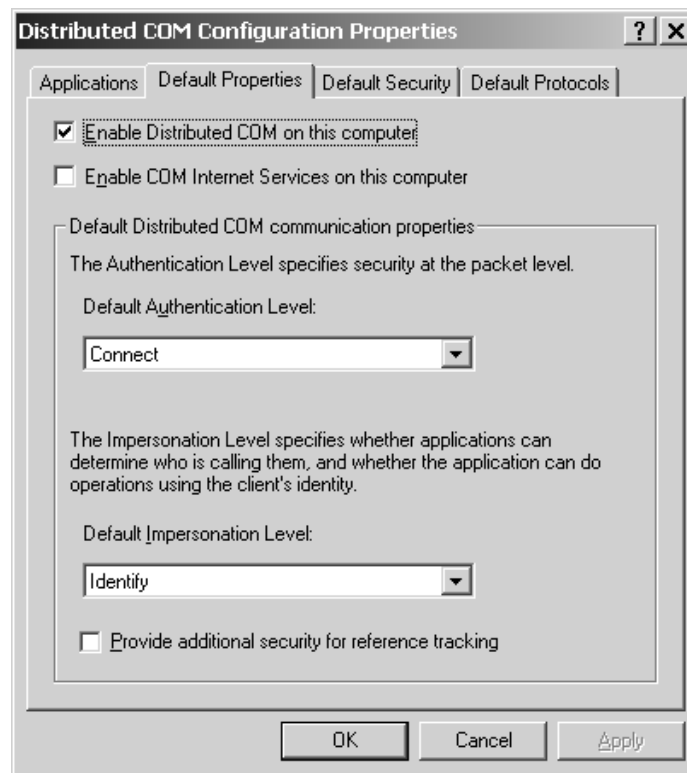
6. Click the **Apply** button to save your settings.

3.3.2 Configuration of DCOM Parameters under Windows XP / Vista / Windows 7

To Set Up General DCOM Parameters

1. Launch the DCOM configuration software as described above (Section 2.2, DCOMCNFG Configuration Software).

   The "Component Services" dialog box is displayed.
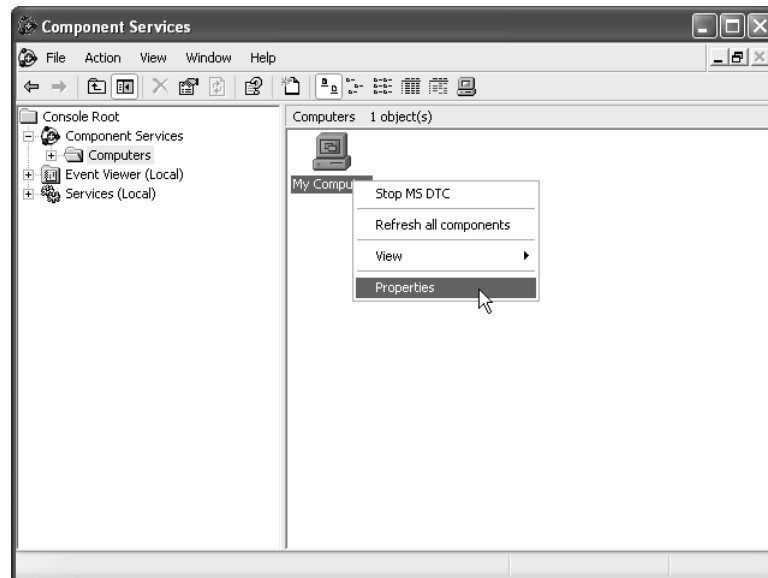
2. In the left-hand list, expand the `Component Services` category and select the `Computers` folder.
3. Right-click the **My** `Computer` icon in the right-hand window pane to open the short-cut menu.
4. Choose **Properties** to open the "My Computer Properties" dialog box.
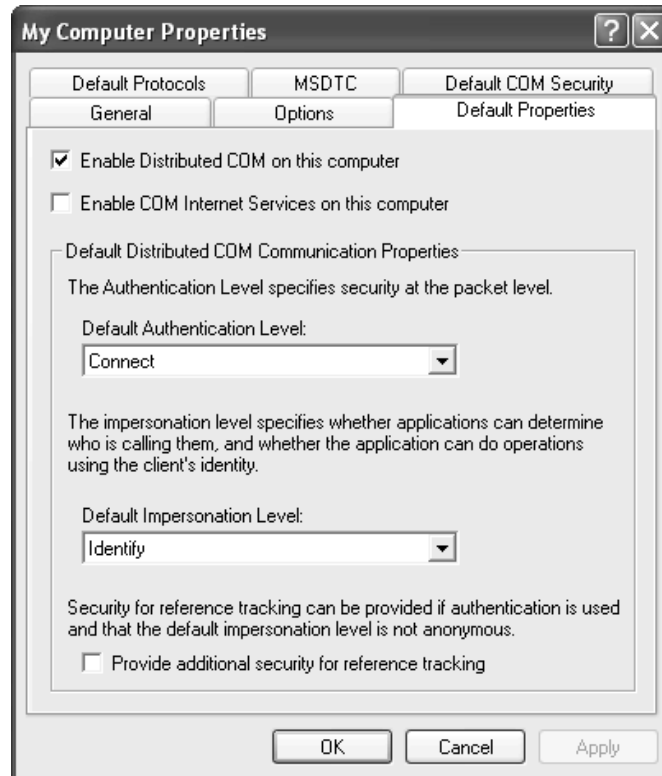
5. Select the **Default Properties** tab.
6. Enable the **Enable Distributed COM on this computer** option.
7. Set the value of the "Default Authentication Level" option to `Connect.`
8. Set the value of the "Default Impersonation Level" option to `Identify.`
9. Click **OK** to save your settings.

   The "My Computer Properties" dialog box is closed.

3.4 INCA Server Configuration

Generally, for the test bench connection under ASAM MCD-3MC, INCA is addressed as a server application. The INCA server requires the INCA software Release 5.0 or higher. The user option to use the new ASAM MCD-3MC standard must be enabled.

First the parameters for ASAM MCD-3MC are configured, followed by any application-specific parameters, depending on the network environment.

3.4.1 Server Configuration under Windows 2000

To Set Up DCOM Parameters for ASAM MCD-3MC

1. Launch the DCOM configuration software as described above (Section 2.2, DCOMCNFG Configuration Software).
2. Select the **Applications** tab.
3. In the "Applications" list box, select the `ASAMMCD3MC4INCA` application.
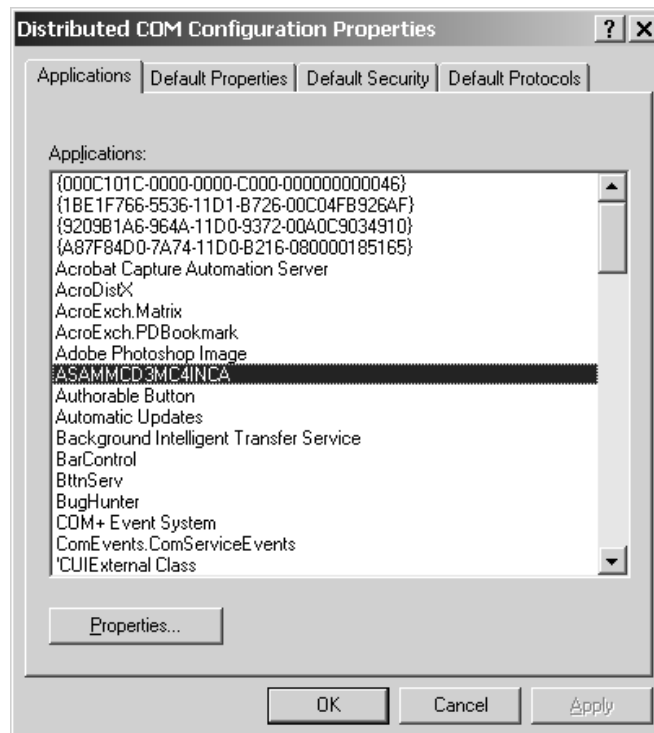
4. Click the **Properties** button.

   The "ASAMMCD3MC4INCA Properties" dialog box is displayed.

5. Select the **Location** tab.
6. Enable the **Run application on this computer** option.
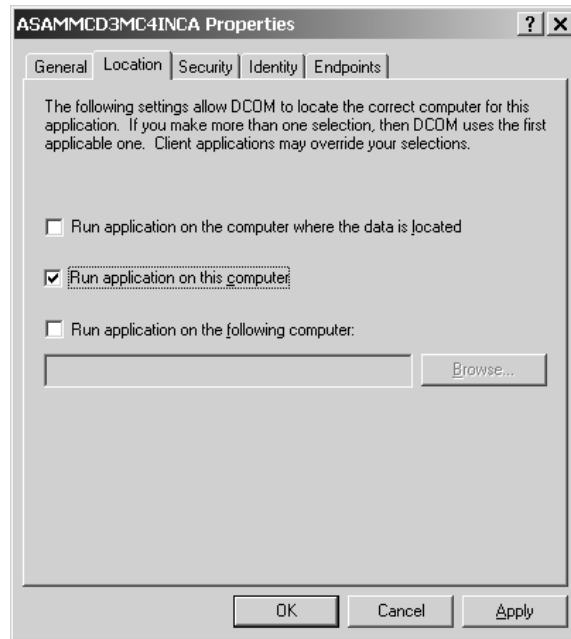
7. Select the **Security** tab.
8. Select the **Customize** option in the "Launch Permissions" section.
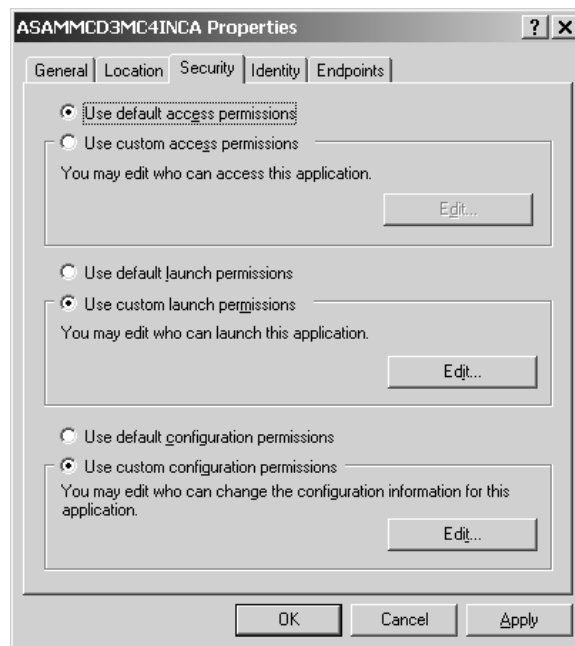
9. Click the **Edit** button to define the launch permissions.

   The "Registry Value Permissions" dialog box is displayed.

10. Make sure that the "Name" list includes the **SYSTEM** user account and the account of the user who will log onto the client computer. For both users, the **Allow launch** access type must be set.

11. Click the **Add** button to enter additional users if required.

12. Click **OK** to apply your settings.

    The "Registry Value Permissions" dialog box is closed.

13. Click the **Apply** button to save your settings.

If you are working in a Windows NT domain or under an Active Directory server, you also must specify the access permissions for the server application. Otherwise, this step concludes the server configuration.

To Specify Access Permission for the Server Application

1. In the "ASAMMCD3MC4INCA Properties" dialog box, select the **Security** tab.
2. Select the **Use custom access permissions** option.

3. Click the **Edit** button to specify the access permissions.

   The "Registry Value Permissions" dialog box is displayed.

4. Make sure that the "Group or user names" list includes the INTERACTIVE and SYSTEM user accounts and the account of the user who will log onto the client computer. For all users, the **Allow access** access type must be set.
5. Click the **Add** button to enter additional users if required.
6. Click **OK** to apply your settings.

   The "Registry Value Permissions" dialog box is closed.

7. Click **OK** to save your settings.

   The "ASAMMCD3MC4INCA Properties" dialog box is closed.

8. Click **OK** to close the "Distributed COM Configuration Properties" dialog box.

   This concludes the server configuration under Windows 2000.

## 3.4.2   Server Configuration under Windows XP / Vista / Windows 7

To Set Up DCOM Parameters for ASAM MCD-3MC

1. Launch the DCOM configuration software as described above (Section 2.2, DCOMCNFG Configuration Software).

   The "Component Services" dialog box is displayed.

2. In the list on the left, expand the `Component Services\Computer\My Computer\DCOM Configuration` categories.

   The right-hand window pane displays a list of registered applications.

3. Right-click the `ASAMMCD3MC4INCA` application in the "Applications" list to open the short-cut menu.

4. Select **Properties** to open the "ASAMMCD3MC4INCA Properties" dialog box.

5. Select the **Location** tab.
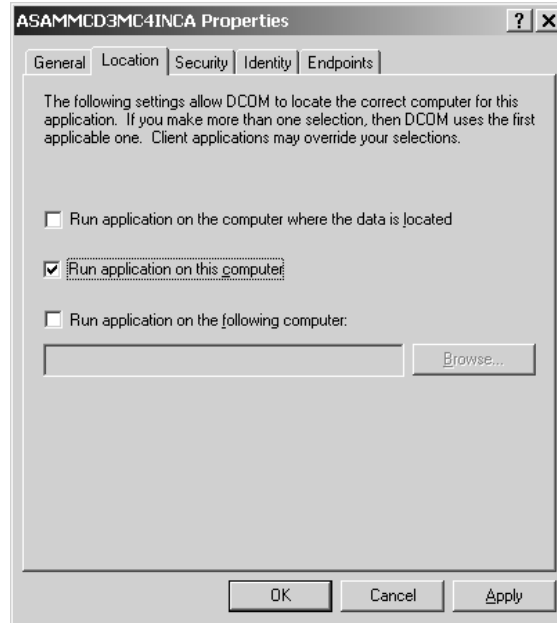
6. Enable the **Run application on this computer** option.
7. Select the **Security** tab.

8. In the "Launch Permissions" group, enable the **Customize** option.
9. Click the **Edit** button to define the launch permissions.

   The "Launch Permissions" dialog box is displayed.
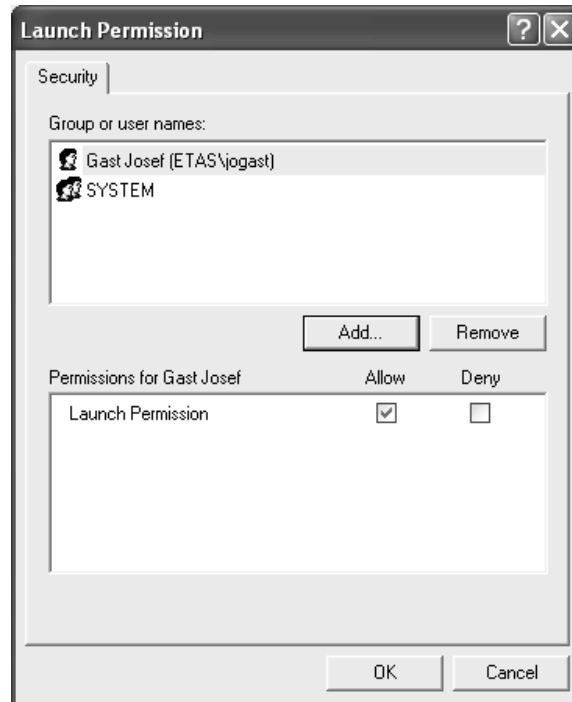
10. Make sure that the "Group or user names" list includes the SYSTEM user account and the account of the user who will log onto the client computer. For both users the "Launch Permission" option must be set to **Allow.**
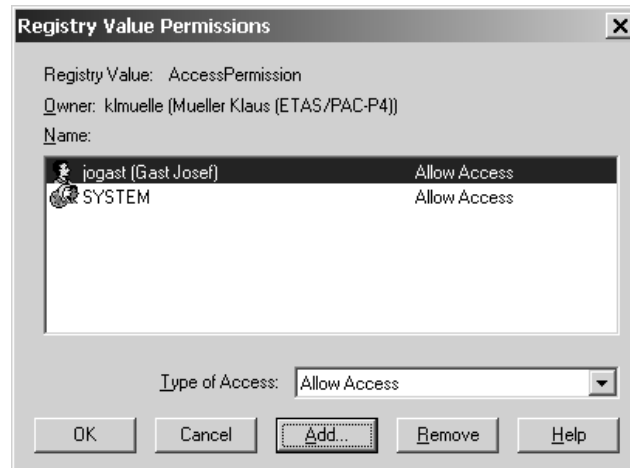11. Click the **Add** button to enter additional users if required.
12. Click **OK** to apply your settings.

    The "Launch Permissions" dialog box is closed.

13. Click the **Apply** button to save your settings.

If you are working in a Windows NT domain or under an Active Directory 2003 server, you also have to specify the access rights for the server application. Otherwise, this step concludes the server configuration.

To Specify Access Permission for the Server Application

1. In the "ASAMMCD3MC4INCA Properties" dialog box, select the **Security** tab.
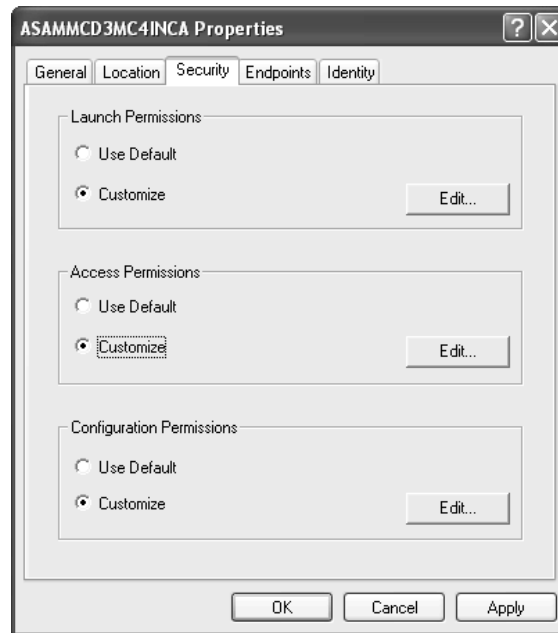
2. In the "Access Permissions" group, enable the **Customize** option.
3. Click the **Edit** button to specify the access permissions.

    The "Access Permission" dialog box is displayed.

4. Make sure that the "Group or user names" list includes the INTERACTIVE and SYSTEM user accounts and the account of the user who will be logging onto the client computer. For both users, the "Access Permission" option must be set to **Allow**.
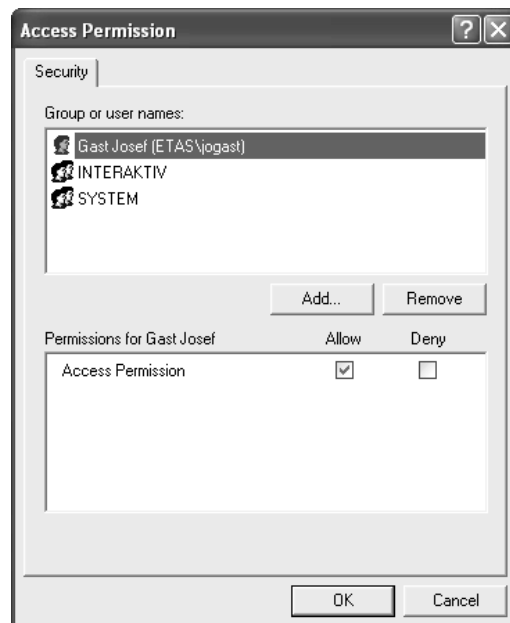
5. Click the **Add** button to enter additional users if required.
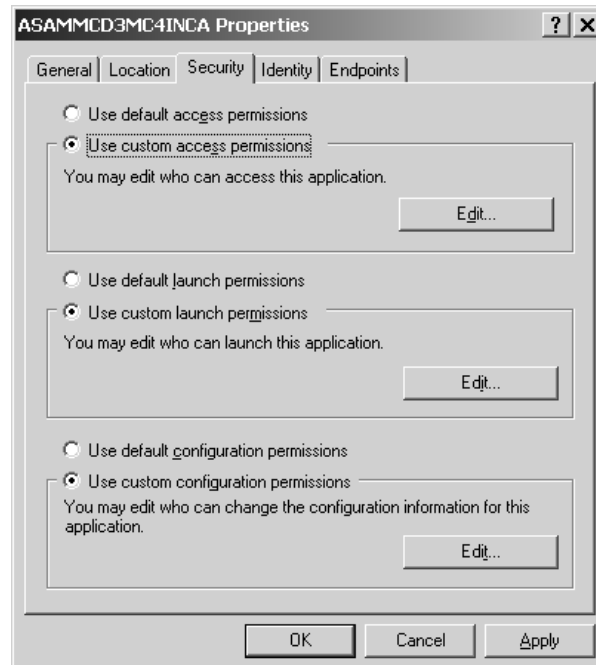
6. Click **OK** to apply your settings.

    The "Access Permissions" dialog box is closed.
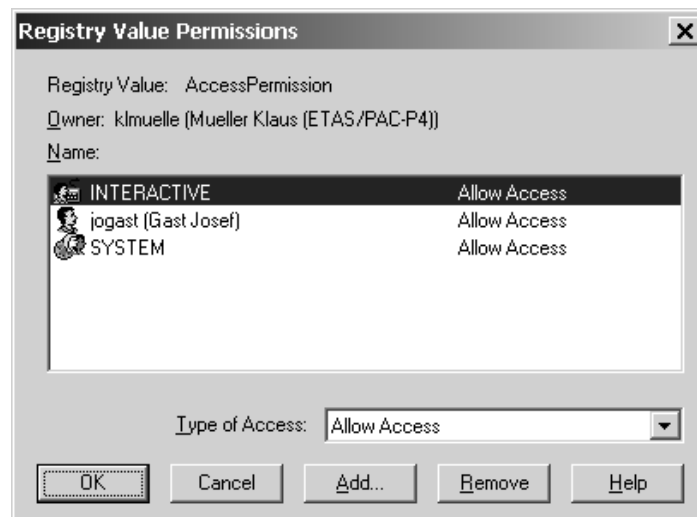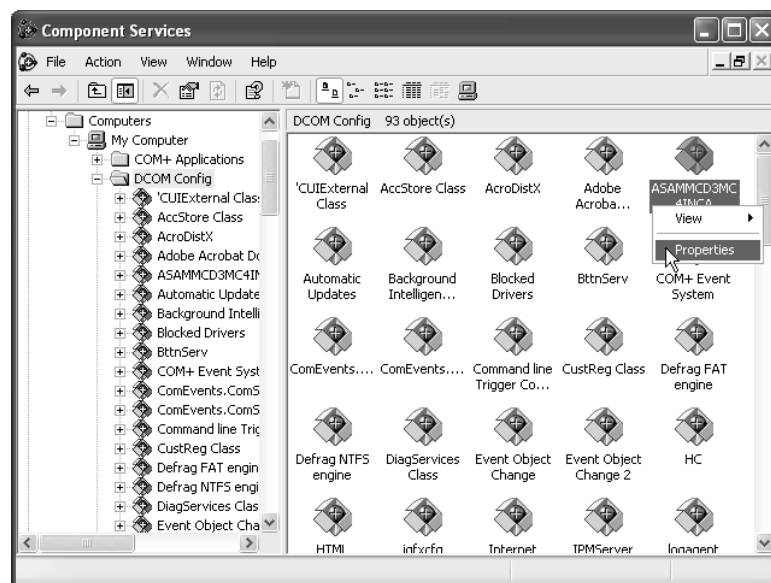
7. Click **OK** to save your settings.
8. The "ASAMMCD3MC4INCA Properties" dialog box is closed.
9. Choose **File** Æ **Exit** to close the "Component Services" window.

    This concludes the server configuration under Windows XP / Vista / Windows 7.

3.5 ASAM MCD-3MC Client Configuration

The client computer runs your test bench software that communicates with the INCA server. In general, there is no INCA software installed on the client computer so that first some INCA components must be installed before you can begin the configuration.

After this, the DCOM parameters are installed on the client computer; first the general DCOM settings are set, followed by the application-specific settings.

3.5.1 Preparations

To allow communication with the INCA server, you first need to transfer files from your INCA server computer to your client computer and register them on the client computer.

To Prepare the Client Computer

1. On the server computer, select the `%INCA%\ASAP3\Client Setup\` directory.
2. Copy the entire directory contents into any directory on the client computer.
3. In the target directory, execute the `SETUP.BAT` file to register the ASAM MCD-3MC components on the computer.

In addition, the client program needs to be registered on the client computer as an application to allow setting the application specific DCOM parameters.

Registering the Visual Basic Client

If you are using a client application created under Visual Basic, you first must register Visual Basic itself as an application on the client computer. To register, first generate an application ID and enter it into the Windows Registry.

To Generate an Application ID

1. Use the `GUIDGEN.EXE` program to generate a unique ID for the application.

In the following set of instructions, you will use the generated ID `%GUID%` to register Visual Basic.

To Register Visual Basic

1. Choose the **Run** option in the **Start** menu.

    The "Run" dialog box is displayed.

2. Type in the `REGEDIT` command and click **OK.**
3. The "Registration Editor" dialog box is displayed. This window is used to create the registration keys for your application.
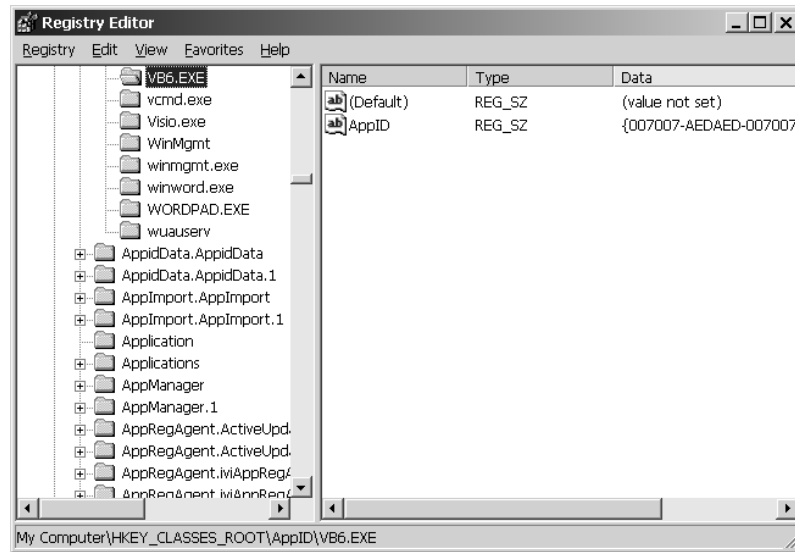
4. Create the following sequential entries in the `HKEY_CLASSES_ROOT` (HKCR) directory of the registration editor:

Path	Name	Type	Value
HKCR\AppID	VB6.EXE	Key	
HKCR\AppID\VB6.EXE	AppID	String	%GUID%
HKCR\AppID	%GUID%	Key	Visual Basic 6
HKCR\CLSID	%GUID%	Key	Visual Basic 6
HKCSR\CLSID\{%GUID%}	LocalServer32	Key	Full path to `VB6.EXE` file

Tab. 2-1   Registration editor

5. First select the specified path and then choose the **Edit ® New ® ‹ TYPE ›** menu option.
6. Then enter the name of the entry and its value, if applicable, in the corresponding dialog windows.
7. Choose the **Registry ® Exit** menu option to close the Registry **Editor**.

    This concludes the registration of Visual Basic, and you can set the appropriate DCOM parameters.

3.5.2   Client Configuration under Windows 2000

To Set Up DCOM Parameters for ASAM MCD-3MC

1. Launch the DCOM configuration software as described above (Section 2.2, DCOMCNFG Configuration Software).
2. Select the **Applications** tab.
3. In the "Applications" list box, select the `ASAMMCD3MC4INCA` application.

4. Click the **Properties** button.

   The "ASAMMCD3MC4INCA Properties" dialog box is displayed.

5. Select the **Location** tab.
6. Enable the **Run application on the following computer** option.

7. Enter the name of the server computer or choose **Browse** to select the name of the server computer in a dialog box.
8. Click **OK** to save your settings.

   The "ASAMMCD3MC4INCA Properties" dialog box is closed.

> ⓘ **NOTE**
>
> If you have installed two network cards each on the INCA server and the 3MC client, one for your company network and the other for accessing the measurement devices, instead of the name, you can specify the IP address of the network card connecting the INCA server to the measurement devices. This will give you a much-improved data throughput.

End points are used for communicating between the server and the client application. To allow this, the client application must be configured so that the user logged on at the server computer is able to access it.

To Specify Access Permissions for the Client Application

1. Launch the DCOM configuration software as described above (Section 2.2, DCOMCNFG Configuration Software).
2. Select the **Applications** tab.

3. In the "Applications" list box, select your client application.

   In the following set of instructions, the Visual Basic 6 application already registered above will be configured as an example.
4. Click the **Properties** button.

   The "Visual Basic 6 Properties" dialog box is displayed.
5. Select the **Security** tab.
6. Select the **Use custom access permissions** option.
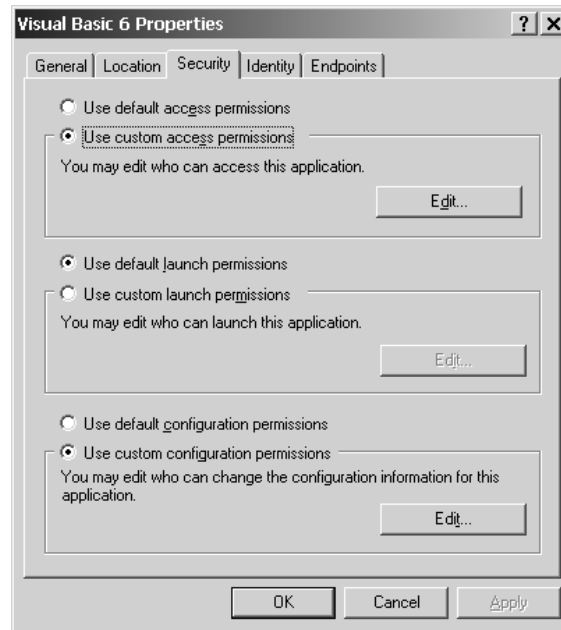
7. Click the **Edit** button to specify the access permissions.

   The "Registry Value Permissions" dialog box is displayed.
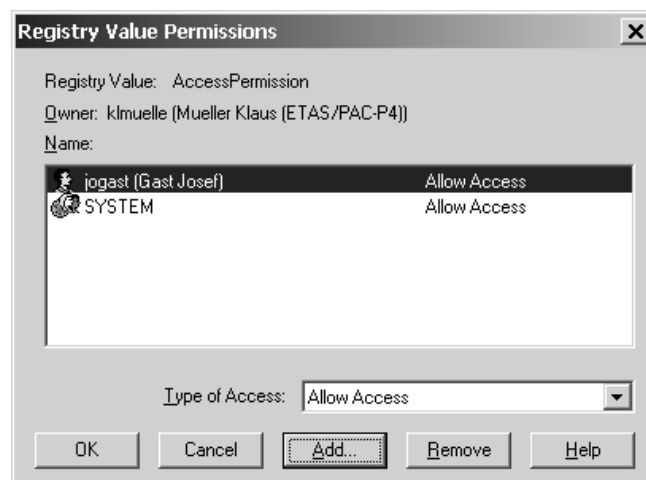
8. Make sure that the "Name" list includes the **SYSTEM** user account and the account of the user who will log onto the server computer. For both users, the `Allow access` type must be set.

9. Click the **Add** button to enter additional users if required.

10. Click **OK** to apply your settings.

    The "Registry Value Permissions" dialog box is closed.

11. Click **OK** to save your settings.

    The "Visual Basic 6 Properties" dialog box is closed. This concludes the client configuration under Windows 2000.

3.5.3 Client Configuration under Windows XP / Vista / Windows 7

To Set Up DCOM Parameters for ASAM MCD-3MC

1. Launch the DCOM configuration software as described above (Section 2.2, DCOMCNFG Configuration Software).

   The "Component Services" dialog box is displayed.

2. In the list on the left, expand the `Component Services\Computer\My Computer\DCOM Configuration` categories.

   The right-hand window pane displays a list of registered applications.

3. Right-click the `ASAMMCD3MC4INCA` application in the "Applications" list to open the short-cut menu.

4. Select Properties to open the "ASAMMCD3MC4INCA Properties" dialog box.

5. Select the **Location** tab.

6. Enable the **Run application on the following computer** option.
7. Enter the name of the server computer or choose **Browse** to select the name of the server computer in a dialog box.
8. Click **OK** to save your settings.

   The "ASAMMCD3MC4INCA Properties" dialog box is closed.

> ⓘ **NOTE**
>
> If you have installed two network cards each on the INCA server and the 3MC client, one for your company network and the other for accessing the measurement devices, instead of the name, you can specify the IP address of the network card connecting the INCA server to the measurement devices. This will give you a much-improved data throughput.

End points are used for communicating between the server and the client application. To allow this, the client application must be configured so that the user logged on at the server computer is able to access it.

To Specify Access Permissions for the Client Application

1. Launch the DCOM configuration software as described above (Section 2.2, DCOMCNFG Configuration Software).

   The "Component Services" dialog box is displayed.

2. In the list on the left, expand the `Component Services\Computer\ My Computer\DCOM Configuration` categories.

   The right-hand window pane displays a list of registered applications.

3. Right-click your client application in the "DCOM Configuration" list to open the short-cut menu.

   In the following set of instructions, the Visual Basic 6 application already registered above will be configured as an example.

4. Select **Properties** to open the "Visual Basic 6 Properties" dialog box.
5. Select the **Security** tab.
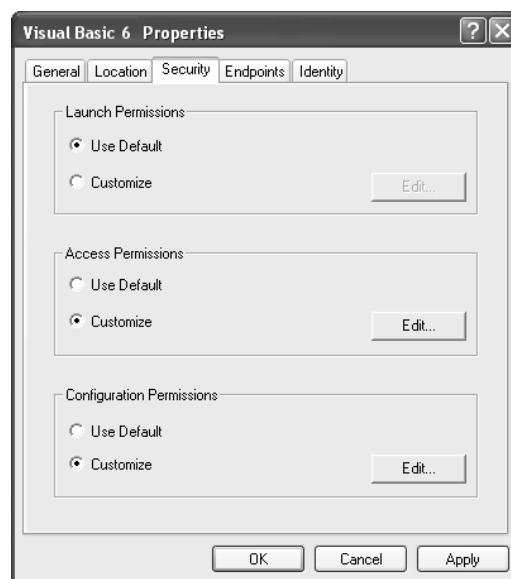
6. In the "Access Permissions" group, enable the **Customize** option.
7. Click the **Edit** button to specify the access permissions.

    The "Access Permissions" dialog box is displayed.
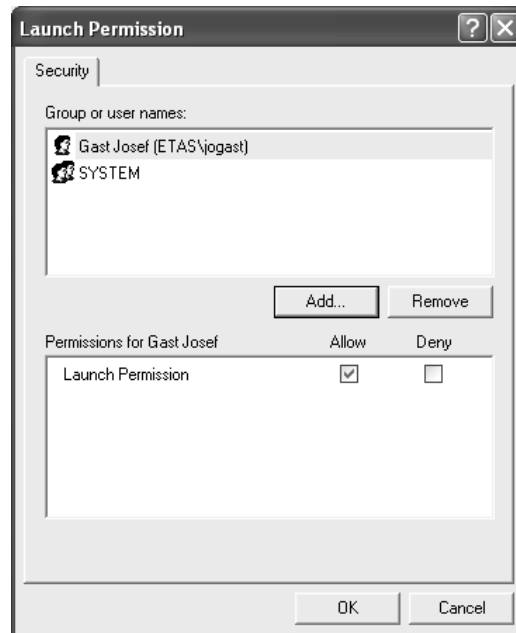
8. Make sure that the "Group or user names" list includes the SYSTEM user account and the account of the user who will log onto the server computer. For both users the "Access Permission" option must be set to "Allow."
9. Click the **Add** button to enter additional users if required.
10. Click **OK** to apply your settings.

    The "Access Permission" dialog box is closed.

11. Click **OK** to save your settings.

    The "Visual Basic 6 Properties" dialog box is closed. This concludes the client configuration under Windows XP / Vista / Windows 7.

3.6 Peer-To-Peer Connections

If you are running a test bench connection based on a peer-to-peer connection, the same user must be logged on at both computers. In addition, the user must be a member of the "Main user" user group.

Furthermore, under Windows XP, the simple form of common file access must be disabled on the client computer to support the end points.

To Disable Common File Access on the XP Client

1. Start the Windows Explorer.
2. Choose the **Tools ® Folder options** menu option.

    The "Folder Options" dialog box is displayed.
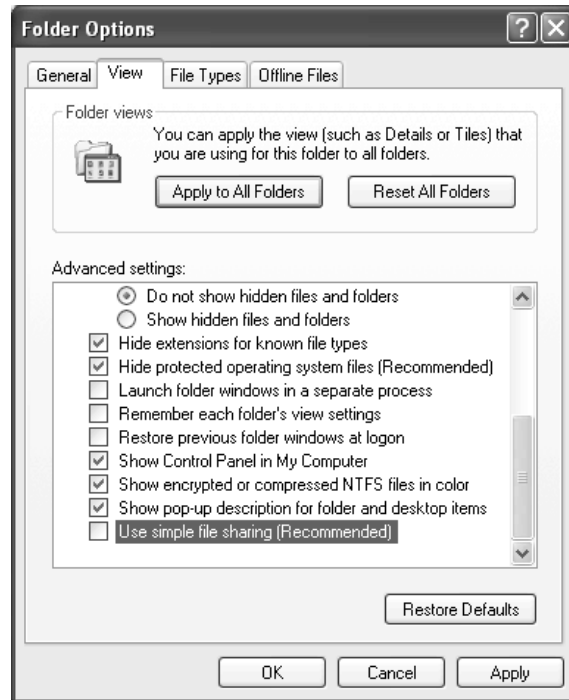
3. Select the **View** tab.
4. In the "Advanced Settings" list, disable the **Use simple file sharing (recommended)** option.

   Click **OK** to apply your settings and to close the "Folder Options" dialog box.

4 Implementation Specific Details

4.1 General

ASAP3.EXE is the COM implementation of an ASAM MCD-3MC server, that is, it provides the interface and the functionality of the M and C stereotypes of the generic ASAM MCD-3MC object model.

The implementation of ASAM MCD 3MC 4 INCA does assume a well behaving COM client. That means, all references to an object are released before the object itself is removed from the collections (or in case of the MCProject object: All references are released before the project is deselected). Since child objects hold references to the parent objects, also child objects must be released before a parent object is removed from a collection. If a client does not follow this rule, chances are high that following steps create errors, e.g., creating an object that uses the same resources as the previous one. Reason: The original object is still living, as long as any client holds a reference to it. Since the object is still living, it still uses the resources within ASAM MCD 3MC 4 INCA and within INCA. This, in turn, can prevent the creation of a new object that uses the same resources.

4.2 Runtime Objects

4.2.1 Management Objects

4.2.1.1 MCSystem

ProgId is ASAMMCD3MC4INCA.MCSystem, which points to the latest installed version of INCA. To address specific INCA installation, use following ProgIDs:

INCA V7.1

- ProgID ASAMMCD3MC4INCA2.MCSystem.INCA7.1
- CLSID {7B2FB1F7-486B-4639-B7C9-A22316FF59CF}.

INCA V7.2

- ProgID ASAMMCD3MC4INCA2.MCSystem.INCA7.2
- *CLSID* {631CB03A-E7E6-45F2-896B-82DFBED160A2}.

CLSID is {A4FA01E0-108F-4AE2-B0CC-6250352250FD}.

The *ShortName* and *LongName* property value is "ASAM MCD 3MC 4 INCA".

Only one instance is supported. An attempt to create a second instance will result in an error. Unfortunately, this specific error can only be seen in the ASAM MCD-3MC log, the client only receives a very generic error, like "Object could not be created".

No multi-client support.

ServerType is eMC.

The *ASAMVersion* property reports currently 1.0.1 (= 1.0 MR01).

NoOfMaxClients is always 1.

The *Version* property shows the file version of ASAP3.EXE. All Versions up to and including 2.0.18 always show version number 1.0.0.

The *PhysFormat* is always 1. You can set a different value to the *PhysFormat* property, but this value is ignored.

SelectProject/SelectProjectByName/DeselectProject: All client references to child objects of *MCProject* must be released before a new *MCProject* can be selected. Deselecting a *MCProject* object does not automatically destroy the whole subtree.
Deselecting the curent project will keep the INCA experiment open or close it, depending on the settings "Close Experiment at eond of session", "Keep Experiment open at end of session" or "Put Experiment into same state as before session start". Starting with INCA 7.1, closing the experiment will no longer save its contents to the database; instead, the changes by the MCD3 client are either dismissed or the INCA experiment stays open (depending on the chosen settings), allowing the users to choose whether the changes are saved or dismissed.

The *Lock* and *Unlock* methods are implemented but only maintain the lock state, they do not actively perform any locking activities (besides sending an event).

4.2.1.2   MCVersion

No special INCA specific implementation details available.

4.2.1.3   MCProject

LongName is not supported and returns **eSYSTEM_METHOD_NOT_SUPPORTED**.

Description is not supported and returns **eSYSTEM_METHOD_NOT_SUPPORTED**.

ActiveDbVehicleInformation returns already the default DbVehicleInformation, even if no one was explicitly selected. This is meant as a convenience functionality for faster setups.

DbProject has no implementation specific details available.

LogicalLinks has no implementation specific details available.

DeselectVehicleInformation: Since only one DbVehicleInformation object is supported, deselecting does not make much sense.

SelectDbVehicleInformation has no implementation specific details available.

SelectDbVehicleInformationByName supports the name "Default" (case insensitive) and will return in this case the default DbVehicleInformation, which is the only available DbVehicleInformation with the original name "VehicleInformation1".

4.2.1.4   MCLogicalLink(s)

MemoryPage supports the pages 0 (reference page) and 1 (working page).

Characteristics has no implementation specific details available.

Collectors has no implementation specific details available.

DbObject has no implementation specific details available.

LockState has no implementation specific details available.

State has no implementation specific details available.

UsedBinary returns the currently used DbBinary object. Currently, it is possible that the DbBinary object does not reflect the correct info of the used database item in INCA, because INCA creates on some operations (upload, download, import) in certain cases (options of ASAP3.EXE) new data sets. Sincd the ASAM MCD-3MC database is not meant to be changed in state ePROJECT_SELECTED, the database and the INCA objects do differ in these cases. Nevertheless, a export of the binary uses the currently used binary in INCA, not the binary with the name in the DbBinary object returned by UsedBinary.

SaveBinaryToFile creates a hex file with the data set of the working page and the code into a file with the given name.

ConnectToModule / DisconnectFromModule: All logical links act synchronized regarding the online state. The first call to of any logical link causes INCA to switch to the online state (measurement and calibration access is switched on). This in turn causes all other logical links to also switch to the state **eONLINE_IDLE**. This is called an implicit online switch. If a Link was switched to **eONLINE_IDLE** implicitely, it is allowed for the client to switch to this state again explicitly. This allows clients that are not aware of the synchronization to work correctly. The same behaviour is true for the switch to **eONLINE_RUNNING** and back to **eONLINE_IDLE** and **eOFFLINE**.

Lock does not perform any actual locking operation, since only one client is possible by design (only one reference to *MCSystem* allowed). It does change the *LockState* to ELOCKED_BY_THIS_OBJECT and does send an *OnLinkLocked* event.

Unlock does not perform any actual unlock operation, since only one client is possible by design (only one reference to *MCSystem* allowed). It does change the *LockState* to ELOCKED_BY_THIS_OBJECT and does send an *OnLinkUnlocked* event.

Name has no implementation specific details.

LongName has no implementation specific details.

Type has no implementation specific details.

RateInfos returns the rate infos for the current logical link. This information stems from the a2l file or from the device's capabilities (especially for measuremet devices). CAN monitoring devices have an additional rate with the code ECSU_FRAME_AVAILABLE. Measurement-only devices and the CalcDev device have an additional rate with the code ECSU_NEW_VALUE.

4.2.1.5   MCRateInfo(s)

The MCRateInfo objects contain the codes from the A2L file or from the device definition. For non-time-based rate infos (e.g., ECSU_FRAME_AVAILABLE, ECSU_NEW_VALUE), the *Value* is 1, for segment synchronous codes, the Value is 3.

4.2.1.6   MCError

There are no implementation specific details.

4.2.1.7   MCValue

There are no implementation specific details.

4.2.2   Calibration objects

As already mentioned in the end user information the current ASAM MCD-3MC implementation supports Scalars, Curves and Maps.For curves and maps the components can be accessed separately (X-/Y-Axes and Curve values in form of vectors and the Map values in form of matrices). This is the reason why we have classes for Scalars, Curves, Maps, Vectors and Matrices as well as a class for the collection of all characteristics.

In the next picture you can see the mapping of the ASAM MCD-3MC characteristic classes to the INCA COM classes:
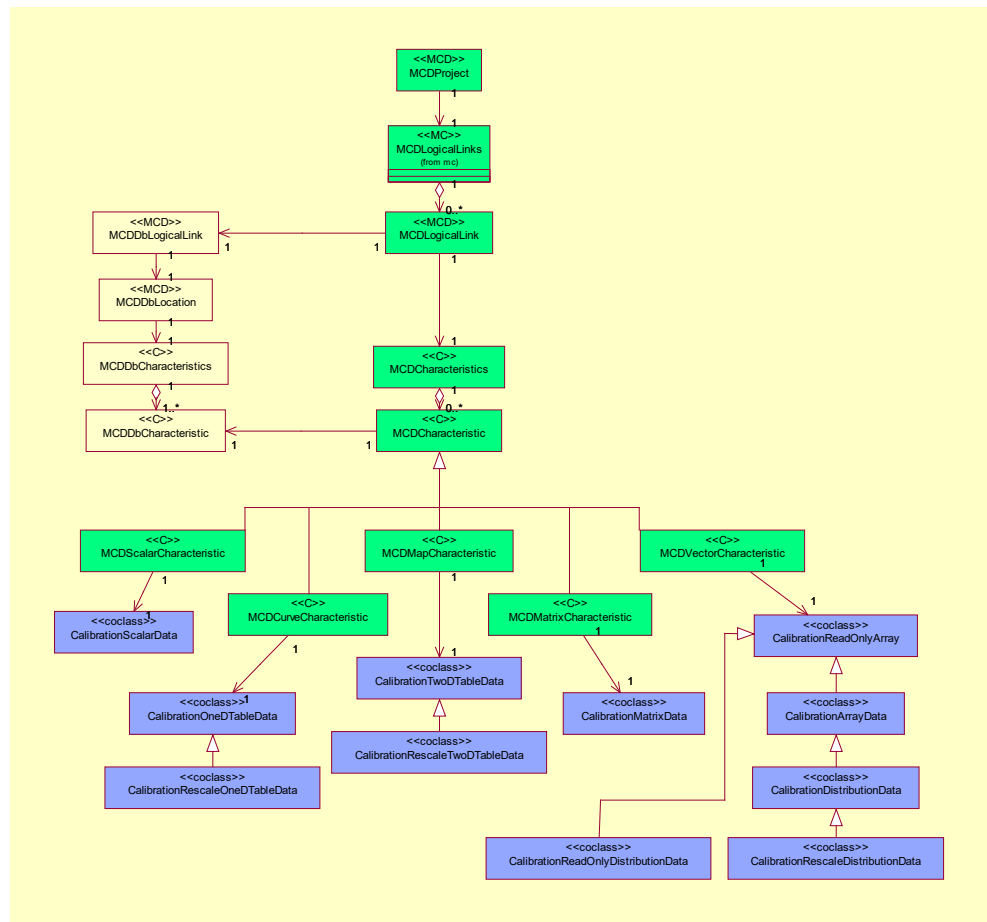
Fig. 3-1   Mapping of ASAM MCD-3MC

Note for FULI users:

If used on a FULI logical link, characteristic objects must be treated in a special way to guarantee that INCA/MCD3-Server are in sync with the FULI hardware:

For a FULI logical link, all characteristics must be created before the logical link is switched online (ConnectToModule is called). After that step, the call to ConnectToModule is required, which will configure the FULI hardware and guarantee synchronized configurations in INCA/MCD3-Server and FULI hardware.

To delete a characteristic on a FULI logical link, it is important to perform an (offline)/online tranisition afterwards, so that again INCA/MCD3-Server and FULI hardware configurations get synchronized. Otherwise, it would e.g., not be possible to create the deleted characteristic again.

4.2.2.1 MCScalarCharacteristic

The method „Read()" is not supported and returns a corresponding error message.

The method "ReadVariant()" will be used by the client to get parameter values from INCA in form of a variant. With the first parameter of the method the user can control if the return value which is kept in the second parameter is a physical or an ECU value. Depending on the data type respectively the representation type of the value which must be read from INCA, different interface functions of the INCA COM API will be called to get the correct data

The method "Write()" is not supported and returns a corresponding error message.

The method "WriteVariant()" will be used by the client to change parameter values in INCA. With the second parameter of the method the client can determine if the value which is kept in the first parameter must be interpreted as an absolute value or if it must be regarded as an offset which has to be added respectively subtracted to/from the current value in INCA. The third parameter is used to control if the value which must be set shall be treated as a physical or an ECU value. Depending on the data type respectively the representation type of the value which must be written to INCA, different interface functions of the INCA COM API will be called to set the correct data.

Designation ASAM MCD-3MC	Designation INCA COM API
MCScalarCharacteristic	CalibrationScalarData_Dispatch

Tab. 3-1   MCScalarCharacteristic

4.2.2.2 MCCurveCharacteristic

With the method "get_Axis()" the client can access the axis of the curve to be able to read or write these values. An interface pointer to a VectorCharacteristic object will be returned (for a detailed explanation concerning the operations which can be performed with this object please refer to the class MCVectorCharacteristic).

With the method "get_Value()" the client can access the curve values itself to be able to read or write these values. Also, an interface pointer to a VectorCharacteristic object will be returned (for a detailed explanation

concerning the operations which can be performed with this object please refer to the class MCVectorCharacteristic).

The method "Read()" enables the client to read both, the axis values of the curve and the curve values itself. The client can define a range (by a start/stop index) where the values of the axis and the curve shall be taken from (for a detailed definition of the indexing please refer to chapter 6.2 of the ASAM MCD-3MC Interface Specification). The values will be returned in the following format:

```
[
  [X(0), X(1), ... , X(N-1)],
  [Z(0), Z(1), ... , Z(N-1)]
]
```

The method "Write()" can be used by the client to write both, the axis values of the curve and the curve values itself. The client can define a range (by a start/stop index) where the values of the curve shall be written to (for a detailed definition of the indexing please refer to chapter 6.2 of the ASAM MCD3 Interface Specification). The values will be expected in the same format as for the "Read()" command (see above).

Designation ASAM MCD-3MC	Designation INCA COM API
MCCurveCharacteristic	CalibrationOneDTableData_Dispatch CalibrationRescaleOneDTableData_Dispatch

Tab. 3-2   MCCurveCharacteristic

4.2.2.3   MCMapCharacteristic

With the methods "get_XAxis()" respectively "get_YAxis" the client can access the X and Y axes of the map to be able to read or write these values. An interface pointer to a VectorCharacteristic object will be returned (for a detailed explanation concerning the operations which can be performed with this object please refer to the class MCVectorCharacteristic).

With the method "get_Value()" the client can access the map values itself to be able to read or write it. An interface pointer to a MatrixCharacteristic object will be returned (for a detailed explanation concerning the operations which can be performed with this object please refer to the class MCMatrixCharacteristic).

The method "Read()" enables the client to read both, the axes values of the map and the map values itself. The client can define a range (by X/Y start/stop indices) where the values of the axes and the map shall be taken from (for a detailed definition of the indexing please refer to chapter 6.2 of the ASAM MCD3 Interface Specification). The values will be returned in the following format:

```
[
  [X(0), X(1), ... , X(N-1)],
  [Y(0), Y(1), ... , Y(M-1)]
  [
    [Z(0,0), Z(1,0), ... , Z(N-1,0)],
    [Z(0,1), Z(1,1), ... , Z(N-1,1)],
             ...
    [Z(0,M-1), Z(1,M-1), ... , Z(N-1,M-1)]
```

```
  ]
]
```

The method "Write()" can be used by the client to write both, the axes values of the map and the map values itself. The client can define a range (by X/Y start/stop indices) where the values of the axes and the map shall be written to (for a detailed definition of the indexing please refer to chapter 6.2 of the ASAM MCD-3MC Interface Specification). The values will be expected in the same format as for the "Read" command (see above).

If in the write command more or less values are defined by the client than expected because of the defined range an error message will be displayed and the values will not be written.

Designation ASAM MCD-3MC	Designation INCA COM API
MCMapCharacteristic	CalibrationTwoDTableData_Dispatch CalibrationRescaleTwoDTableData_Dispatch

Tab. 3-3   MCMapCharacteristic

4.2.2.4   MCVectorCharacteristic

This class is used to simplify the access to all one-dimensional objects (array values, curve values or axes values of a curve or map) in INCA.

The method "Read()" enables the client to get the values of one dimensional calibration objects like the axis values of a curve/map or the curve values itself. For a detailed description of the function parameters please refer to the explanation of the class MCCurvCharacteristic. The values will be returned in the following format:

$[X(0), X(1), … , X(N\text{-}1)]$ or
$[Y(0), Y(0), … , Y(N\text{-}1)]$ or
$[Z(0), Z(1), … , Z(N\text{-}1)]$

The method "Write()" can be used by the client to write the values of one dimensional calibration objects. For a detailed description of the function parameters please refer to the explanation of the class MCCurveCharacteristic. The values will be expected in the same format as for the "Read" command (see above).

Designation ASAM MCD-3MC	Designation INCA COM API
MCVectorCharacteristic	CalibrationReadOnlyArray CalibrationArrayData CalibrationReadOnlyDistributionData CalibrationDistributionData CalibrationRescaleDistributionData

Tab. 3-4   MCVectorCharacteristic

4.2.2.5   MCMatrixCharacteristic

This class is used to simplify the access to all two-dimensional objects (map values) in INCA.

The method "Read()" enables the client to read two dimensional calibration objects like the map values itself. For a detailed description of the function

parameters please refer to the explanation of the class MCMapCharacteristic. The values will be returned in the following format:

[
  [Z(0,0), Z(1,0), … , Z(N-1,0)],
  [Z(0,1), Z(1,1), … , Z(N-1,1)],

      …

  [Z(0,M-1), Z(1,M-1), … , Z(N-1,M-1)]
]

The second method "Write()" can be used by the client to write two dimensional calibration objects like the map values itself. For a detailed description of the function parameters please refer to the explanation of the class MCMapCharacteristic. The values will be expected in the same format as for the "Read" command (see above).

Designation ASAM MCD-3MC	Designation INCA COM API
MCMatrixCharacteristic	CalibrationMatrixData_Dispatch

Tab. 3-5   MCMatrixCharacteristic

4.2.2.6   MCCharacteristics

This class must be used by the client to instantiate the wanted characteristic objects (scalars, curves and maps) and keep a reference to it (it acts as a collection of characteristic objects). To do this the method "Add()" has to be used. With the first parameter the data base object of the wanted calibration object must be handed over to be able to instantiate the correct characteristic. In the second parameter the method returns an interface pointer to the instantiated characteristic to enable the client to manipulate it according to his needs.

With the methods "GetItemByName()" and "GetItemByIndex()" it's possible to get access to the wanted characteristic object of the collection which was generated with the "Add()" method before.

When a characteristic object of the collection has to be removed the methods "RemoveByName()" or "RemoveByIndex()" can be called. All characteristics can be removed with the method "RemoveAll()".

4.2.3 Measurement objects

4.2.3.1   MCCollector

In INCA it is possible to configure variables as display variables and/or recording variables. The equivalents on the Collector side are determined by the StorageType that can be either **eST_AUSY** (display varaibles) or ᴇST_FILE (recorder variables). Through a setting in the options, it is furthermore possible to switch display for recorder variables on/off.

The Measurement is started in parallel for all Collectors, i.e., the first Collector that is started starts the measuremnt within INCA. On the other hand, the last stopped Collector leads to a real stop of the measurement within INCA.

If a Collector with a negative StartDelay is activated, the measurement is already started for collection of pretrigger data (this is equal to the pretrigger time in

INCA). In this case no further wiring is allowed. A positive StopDelay equals the posttrigger time in INCA. The user should note that StartDelay and StopDelay allow positive and negative values.

While Measurement is running no further wiring is possible (i.e., Check/Change of other Collectors). A violation of that rule is answerd with an error message that tells the user, that it is impossible to do the current operation while another Collector is in state EOS_ACTIVATED or EOS_STARTED.

A Collector with StorageType EST_FILE can also provide data to a client (EST_AUSY) (see setting in options dialog). In this case the FormatID contains the number of samples as always **e**FID_USER is assumed (EFID_ASCII, EFID_ODS and EFID_ATF are not supported in the current implementation).

Measurement of Characteristics is not supported now.

The minimum update rate for an event Client is 100ms (faster data is always provided in blocks of 100ms). This is a limitation of the TargetServer because the data from the hardware is provided in 100ms blocks. There is no possibility to alter this behaviour.

In most cases every variable can be wired in every available raster. Due to INCA limitations there are some exceptions:

Variable wired 1st time in	Time raster	Same variable wired 2nd time in	Description
Recording	N/A	Recording	**Error, wiring not possible (also the old ASAP3 and INCA GUI do not support such a feature)**
Measurement	=	Recording	**OK**
Recording	=	Measurement	**OK**
Measurement	‹	Recording	**Error, wiring not possible**
Recording	‹	Measurement	**OK (Downsample Measurement from Recording)**
Measurement	›	Recording	**OK (Downsample Measurement from Recording)**
Recording	›	**Measurement**	**Error, wring not possible**

Tab. 3-6   INCA limitations exceptions

That leads to an internal downsampling, i.e., the faster raster is always downsampled to the slower raster, as INCA can only wire one variable into one specific raster.

Downsampling between measurement values with different representation types (`eRT_ECU/eRT_PHYSICAL`) is not possible.

CalcDev variables must be wired with raster NEW_VALUE.

CanMonitoring variables must be wired with raster FRAME_AVAILABLE. This allows the usage of different CAN frames in one Collector.

The same variable can be wired in different equidistant rasters, but only exclusive in equidistant and synchro rasters, because downsampling between these two raster types is simply not possible.

With FULI LogicalLinks, the wiring rules are slightly different, as shown in the following tables (①, ② = sequence, order of execution):

Wiring rules for FULI (Add):

A (Norm)	A (FULI)	Action
① Measurement		① wire normal
① Measurement	② Measurement	① wire normal, ② enable FULI
② Measurement	① Measurement	① wire normal, enable FULI
	① Measurement	① wire normal, enable FULI

Tab. 3-7   Wiring rules for FULI (Add)

Wiring rules for FULI (Remove):

A (Norm)	A (FULI)	Action
① Measurement		① remove
① Measurement	② Measurement	② disable FULI, remove
② Measurement	① Measurement	① disable FULI, ② remove
	① Measurement	① disable FULI, remove

Tab. 3-8   Wiring rules for FULI (Remove)

If a variable is wired in a Collector from a FULI-LogicalLink with 100ms and afterwards the corresponding LogicalLink (w/o FULI) with 10ms, the EtherCAT data will arrive with 10ms!

For all Collectors with FULI variables, no data is transmitted to the client!

Recording is not supported for FULI LogicalLinks.

The Check method of the Collector does not return immediatelly in case of an error but continues the wiring through all CollectedObjects. At the end a summary of all errors that happened during the wiring process is returned to the Client.

The following Collector related settings can be influenced by the Settings class:

Settings.MCDStorageType = EST_FILE or EST_AUSY (default storage type)
Settings.LimitVariablesPerWindow = True/False (turn on/off the limitation of variables in one Window)
Settings.VariablesPerWindow = Value (number of variables that will be displayed together in one window; Settings.LimitVariablesPerWindow must be set to True)

4.2.3.2 MCBuffer

The Buffer is configured with the following default values:

Downsampling = 1
Rate = 100ms (Unit = **eCSU_MS**, Value = 100)

Size = 1000
TimeStamping = True

For faster devices it might be helpful to increase the size of the buffer if overflows occur, but this is not always a practicable solution, because the system and the client / server connection must be fast enough to transport the data. But if the overflows happen only in special situations (i.e., high system workload due to some short operations) it makes sense to increase the buffer size to catch up those data peaks.

Hint for polling mode: As in polling mode the client has no error handler connected, he will be noticed about overflow errors only by the error member of the MCResult object. By calling MCResult::HasError he can check if there has been an error during measurement of this an the last line, e.g. overflow events. In case of a buffer overflow the recognition at the client is always delayed. Therefore, it is not recommended to increase the buffer size beyond some limits. If a buffer has a size of 5000 lines and an overflow occurs, the client will first have to consume 5000 lines before he gets the line that signals him the overflow error that happened in the past.

It is possible to change the default values for every new Buffer through the Options dialog. Additionally, it is possible to change the above default values for all new Buffers through the Settings class:

Settings.MCDBufferDownSampling = Value (1... INT_MAX)
Settings.MCDBufferRate = Value (interpreted together with unit)
Settings.MCDBufferRateUnit = Unit (One of the eCSU_xx constants)
Settings.MCDBufferSize = Size (Lines of data in the buffer)
Settings.MCDBufferTimeStamping = True/False (Turn time stamping on/off)

4.2.3.3 MCStorageAusy

This is the default storage type of the system. The default value for NumberOfSamples is 10, i.e., every 10 lines of new data an event is sent to the client, to inform him about the new data. It is possible to change those values through the options dialog and additionally through the Settings class:

Settings.MCDCollectorNumberOfSamles = Value (1... INT_MAX)

4.2.3.4 MCStorageFile

The default value for FormatID is always eFID_USER. Beyond that the user can select an alternative storage format in the Options dialog. The alternative format is turned off by default. Additionally, it is possible to activate the alternative format through the Settings class:

Settings.MCDCollectorStorageFile = Filename to which a Collector of type eST_File stores the data.
Settings.UseRecorderFileFormat = True/False (Turn on/off the alternative

format)

Settings.RecorderFileFormat = Value (Set one of the available formats)

Possible values for the RecorderFileFormat are:

0 = ETAS binary data file format
1 = DIADEM ATF measure format
2 = ASCII
3 = ASCII (Multirate)
4 = Matlab M-File
5 = Matlab M-File (Multirate)
6 = FAMOS (imc)
7 = MDF (Measure Data Format)
8 = MDF4 (Measure Data Format for VSG using 4 bytes floats only)

Note: The list of possible values for the RecorderFileFormat depends on the choosen INCA Primary Recording Format. If MDF4 is choosen as Primary Recording Format, only MDF4 is allowed as value for the RecorderFileFormat.

All Collectors of StorageType eST_FILE use together one file to store the data. It is not possible to have a separate file for every Collector, i.e., setting the file name for one Collector, updates the file names for all other Collectors of StorageType eST_FILE.

Settings.MCDCollectorRealTimeData = True/False (Turn on/off realtime data for a Collector of type eST_FILE. If this value is set to True, the Collector saves the data into a file and additionally provides it to a Client as a Collector of type eST_AUSY.)

Settings.MCDShowRecordedValues = True/False (Turn on/off wether recorded values should be visible in the experiment environment or not.)

4.2.3.5 MCTimeDelay

The time delay is used within a Collector through the StartDelay and the StopDelay. They are preset with the following default values:

StartDelay: 0ms (Unit = ETR_MS, Value = 0)
StopDelay: 0ms (Unit = ETR_MS, Value = 0)

In both cases the resolution is set to eTR_MS and the value as seen above is set to 0. The valid range for the value is from 0 to LONG_MAX.

As for all other default values they can be influenced programatically by the Settings class:

Settings.MCDStartDelay = Value (Value; interpretation depends on unit)
Settings.MCDStartDelayUnit = one of the eTR_xx constants
Settings.MCDStopDelay = Value; interpretation depends on unit)
Settings.MCDStopDelayUnit = one of the eTR_xx constants

4.2.3.6 MCResults Tree

The MCResults tree is created on the client side only (The original values are marshaled by value as a binary data stream). Therefore, it is necessary that the Client releases all COM objects as soon as possbile. See 5.4.1 for more information.

4.2.3.7 MCResult(s)

The Parent property of the MCResults object always returns NULL, because the reference to the Collector is not marshaled to save time. Additionally, it is not recommended to access any server objects outside the MCResults tree within the OnCollectorResults event, because this would cause a delay that might prevent the client from retrieving the results fast enough. This can result in permanent buffer overflows on the server.

4.2.3.8 MCResponse(s)

In the Response class the methods getResponseMessage and getResponseMessageVariant are not supported.

4.2.3.9 MCResponseParameter(s)

In the ResponseParameter class the methods getDataBaseObject, getDecimalPlaces, getRadix, getUnit, getRangeInfo are not supported. To database object can be accessed through the CollectedObjects of a Collector. Through the CollectedObjects it is possible to also access unit and range information.

4.2.3.10 Timestamp in the MCResponseParameter named "TimeStamp"

As described in the ASAM MCD-3MC specification the element with the name "TimeStamp" contains the timestamp. This value is provided as double value in seconds (original TargetServer value). In the standard it is defined as UINT64 in microseconds. A conversion to microseconds would only generate higher numbers, but not more precission. Therefor it was decided to take over the existing TargetServer value.

4.2.4 Database objects

The current implementation provides an implementation for the database objects that are required to provide the functionality of the old ASAP3 protocol.

4.2.5 MCDbProjectDescription(s)

Depending on the option "Search workspaces, description and binary files in all folders of the INCA database", a list of all projects in the INCA database or only a list of all projects in and below the database folder of the current active workspace is provided in this collection. A change of the option has only an effect if the MCSystem object is destroyed and newly created, that is, the current session with the server is closed (if any) and a new session is established. A project name consists of the full path of the database directory, in which the workspace resides, and the workspace name itself. There is no preceding backslash. Example: "DEFAULT\Workspace".

LongName is not implemented, because it is not supported by INCA.

Description is not implemented, because it is not supported by INCA.

4.2.6 MCDbProject

LongName is not implemented, because it is not supported by INCA.

Description is not implemented, because it is not supported by INCA.

Comment and *ProjectNo* are read from / stored into the tagged INCA Workspace comment. The following tags are supported:

*Comment:
*ProjectNo:

They are embedded between a starting

*ASAM MCD 3MC START

and an ending

*ASAM MCD 3MC END

tag.

DbLocations has no implementation specific details available.

DbPhysicalVehicleLinkOrInterfaces has no implementation specific details available.

DbVehicleInformations has no implementation specific details available.

4.2.7 MCDbProjectConfiguration

As with *MCSystem:SelectProject/SelectProjectByName/DeselectProject*, also all client references to the previous project have to be released before the *load* or *add* methods are called.

The *add* method does not yet perform a hardware scan, so no devices are available in this project. They must be added manually within INCA.

4.2.8 MCDbVehicleInformation(s)

This object is only a dummy object. Its sole purpose is to provide access to the DbLogicalLinks and the DbPhysicleVehicleLinkOrInterfaces collections. Because of that, there is only one MCDbVehicleInformation available in the collection. It has the name "VehicleInformation1" but can also be selected by the name "default".

There is no further implementation specific detail available for both collection and object.

4.2.9 MCDbLogicalLink(s)

Logical links are named following this scheme:

<Description File Name>	_	<Device Name>	_	<Protocol Type>
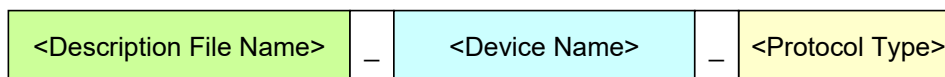
Fig. 3-2   Logical links

Example:

DEFAULT\0400_ETKC:1_ETK

The description file name is the name and path of the description file within the INCA database. It has no preceding backslash. The device name is the name of the device in the INCA Hardware Configuration. The protocol type is "ETK" for ETK devices, "CAN" for the CAN monitoring device, "CCP" for ECUs connected via CCP, "KWP2000" for ECUs connected via KWP2000, "McMess" for devices using

McMess, and GENERIC for all others. See chapter "3.2.10 MCDbPhysicalVehicleLinkOrInterface(s)" for a detailed list of INCA devices and associated protocol types.

LogicalLinks that support the FULI/MCE-Feature have a #FULI appended to their name. In case the option "Always use FULI device if available" is checked, those FULI logical links do not have #FULI appended but replace the normal logical links.

Measurement devices do not have an actual description file, like an a2l file. Instead, they only have an internal description file, which does not show up in the INCA database as a separate item. Therefore, for measurement files the logical link names are built up in a different manner:

device://<Device Name>_Auto	_	<Device Name>	_	<Protocol Type>

Fig. 3-3   Link names in a different manner

Example:

device://ES610 / AD:1_Auto_ES610 / AD:1_GENERIC

The part, where ECU logical links have the description file name, is here replaced with the text "device://<device name>_Auto". The prefix "device://" is an indicator for the fact that this is a measurement device. The next part, "<device name>_Auto" is a substitute for the not available description file. You can think of this as a made-up name for the description file. The next part, "<device name>", is again the actual device name. This is analogous to the ECU devices. Also, the last part, "<protocol type>", is the same syntax as with ECU devices, but since SMB was not added as a protocol in the ASAM standard, "GENERIC" is used for all measurement devices (including those connected via Ethernet or plugged in to an ES1000 system).

The option "Use current workspace configuration only" influences which MCDbLocations and hence which MCDbLogicalLinks are found and listed.

There are no further implementation specific details available for both collection and object.

4.2.10 MCDbPhysicalVehicleLinkOrInterface(s)

The following table shows the `tagMCPHYSICALLINKORINTERFACETYPE` which the *Type* method returns, depending on the underlying INCA device:

INCA Device Type	ASAM MCD-3MC DeviceType	ASAM MCD-3MC Logical Link Extension (_<protocol type>)
ETK-Testdevice	eETK	_ETK
ETKC	eETK	_ETK
CAN-Monitoring	eCAN	_CAN
CCP	eCCP	_CCP
KWP2000	eKWP2000	_KWP2000
McMess	eMCMESS	_MCMESS
AD-Scan	eUART	_UART
BaroScan	eUART	_UART

LA2	eUART	_UART
LA3	eUART	_UART
LA4	eUART	_UART
Thermo-Scan	eUART	_UART
Dual-Scan	eUART	_UART
All other INCA device types	eGENERIC	_GENERIC

Tab. 3-9   `tagMCPHYSICALLINKORINTERFACETYPE`

4.2.11 MCDbBinary(/-ies)

The content of the list of MCDbBinaries is influenced by the option "Skip write protected datasets".

The method *Add* imports a hex file and creates a new data record. The supported file formats are the same as those that the INCA GUI supports:

Format Name	File Extension
Intel Hex	*.hex
Motorola16	*.s37
Motorola24	*.s28
Motorola32	*.s19

Tab. 3-10   Method adds supported file formats

The new database entry is made in the subfolder that is given within the *ShortName*. The folder is created in case it does not yet exist. In case there is already an item with the given ShortName, the item name is changed to be unique.

GetDefault returns the first binary in the list of available binaries as the default.

RemoveAll has no implementation specific details.

RemoveByIndex has no implementation specific details.

RemoveByName has no implementation specific details.

4.2.12 MCDbLocation(s)

MCDbLocation is only loaded from the INCA DB after it is required from the *MCDbLocations* list by the client. This Fills the *MCDbLocation* and the sub-objects.

The option "Use current workspace configuration only" influences which MCDbLocations and hence which MCDbLogicalLinks are found and listed.

The method *Add* reads in an a2l file into the INCA database, at the folder that is given in the ShortName parameter. In case there is already an item with the given ShortName in the INCA Db, the *Add* method fails. The same is true if INCA runs out of Disk Space or if the file does not exist.

RemoveAll has no implementation specific details.

RemoveByIndex has no implementation specific details.

RemoveByName has no implementation specific details.

4.2.13 MCDbCharacteristic(s)

MCDbCharacteristic-derived objects are created along with the *MCDbLocation*, but not all properties are filled. Only the "Name" property is set originally. This allows the client to ask for a list of the names of the objects via the "Names" property at the *MCDbMeasurements* object. The other properties are delay-loaded from the INCA-Db when the item is requested by the client at the *MCDbMeasurements* object.

Items with DataTypes that are not supported in the ASAM MCD-3MC model will not be included in the MCDbCharacteristics collection since they cannot be instantiated..

Not all of the properties are available with the current implementation.

ShortName returns the INCA name of the characteristic.

LongName returns the INCA comment of the characteristic.

Type returns the INCA characteristic type. Currently, the types ᴇCT_CUBOID, ᴇCT_VAL_BLK and ᴇCT_ASCII are not supported.

UpperLimit returns the INCA Upper Bound / Upper Limit value.

LowerLimit returns the INCA Lower Bound / Lower Limit value.

DisplayIdentifier always returns the default value of "" (empty string).

Format always returns the default value of "" (empty string).

Conversion always returns the default value of "" (empty string).

IsReadOnly always returns the default value of FALSE.

ExtLowerLimit always returns the default value of 0.0.

ExtUpperLimit always returns the default value of 0.0.

Address always returns the default value of 0.

MaxDifference always returns the default value of 0.0.

RefDbCompuMethod always returns a NULL pointer.

4.2.14 MCDbScalarCharacteristic

This class does not have any additional implementation specific behaviour in addition to *MCDbCharacteristic*.

4.2.15 MCDbCurveCharacteristic

AxisDescription has no implementation specific details.

4.2.16 MCDbMapCharacteristic

XAxisDescription has no implementation specific details.

YAxisDescription has no implementation specific details.

4.2.17 MCDbAxisDescription

AxisType always returns the default value of ᴇADA_STD.

DepositType always returns the default value of ᴇDT_DIFFERENCE.

ExtLowerLimit always returns the default value of 0.0.

ExtUpperLimit always returns the default value of 0.0.

Format always returns the default value of "" (empty string).

LowerLimit returns the lower limit given in the A2L file.

MaxAxisPoints returns the MaxAxisPoints value given in the A2L file.

RefDbAxisPts always returns the default value of NULL.

RefDbCompuMethod always returns the default value of NULL.

RefDbMeasurement always returns the default value of NULL.

UpperLimit returns the upper limit given in the A2L file.

IsReadOnly always returns the default value of FALSE.

4.2.18 MCDbMeasurement(s)

MCDbMeasurement objects are created along with the MCDbLocation, but not all properties are filled. Only the "*Name*" property is set originally. This allows the client to ask for a list of the names of the objects via the "*Names*" property at the *MCDbMeasurements* object. The other properties are delay-loaded from the INCA-Db when the item is requested by the client at the *MCDbMeasurements* object.

Not all of the properties are available with the current implementation.

ShortName returns the INCA name of the measurement item.

LongName returns the INCA comment of the measurement item.

Accuracy always returns the default value of 0.0.

ArraySize always returns the default value of 0.

DataType returns the data type for the item.

DisplayIdentifier always returns the default value of "" (empty string).

Format always returns the default value of "" (empty string).

LowerLimit returns the Lower Bound value of INCA.

RefDbCompuMethod

Resolution always returns the default value of 0.

UpperLimit returns the Upper Bound value of INCA.

4.2.19 MCDbCompuMethod(s)

This class is not supported.

4.2.20 MCDbCompuTab(s)

This class is not supported.

4.2.21 MCDbCompuTabBase

This class is not supported.

4.2.22 MCDbCompuVTab(s)

This class is not supported.

4.2.23 MCDbCompuVTabRange(s)

This class is not supported.

4.2.24 MCDbFunction(s)

This class is not supported.

4.2.25 MCDbGroup(s)

This class is not supported.

4.2.26 MCDbModPar

This class is not supported.

4.2.27 MCDbTab(s)

This class is not supported.

4.2.28 MCDbUnit(s)

This class is not supported.

4.2.29 MCDbVTab(s)

This class is not supported.

4.2.30 MCDbVTabRange(s)

This class is not supported.

5 Miscellaneus

5.1 Start/Stop Behaviour

ASAP3.EXE can be started either via the INCA user interface or via the client. In boht cases, the INCA UI is locked in user input. If the ASAP3.EXE and INCA.EXE do not show any GUI when started via the client, the DCOM setup is not correct. Please refer the document "INCA 5.1 ASAM MCD-3MC DCOM Configuration.doc" for further information about proper DCOM configuration.

If a second client tries to connect to INCA via ASAM MCD-3MC by creating an instance of the MCSystem object, the CoCreateInstance call (or any equivalent call in the client) fails. The same happens if there is a reference count problem in the client that causes some dangling references in the server. This situation can be ended by manually closing the ASAP3.EXE GUI and affirmatively respond to the message box, that will appear during shutdown of ASAP3.EXE. In some rare cases, if ASAP3.EXE is currently waiting for a call to INCA to return, even closing the GUI does not help. Here, it is required to terminate the process via the Task Manager.

5.2 Settings

Available also via COM: ASAMMCD3MC4INCA.Settings. This object provides access to the settings in the ASAM MCD 3MC 4 INCA settings dialogs. Not all settings have immediate impact. E.g., the search for projects in whole database must be set before the MCSystem object is instanciated.

All settings are implemented as read/write properties, except where noted otherwise. The measurement related settings are also described in chapter "3.2.3 Measurement objects".

Property Type and Name	Description
BOOL SaveModifiedExperimentAtStartup	If enabled, the INCA experiment is saved at session start (when a project is selected in MCSystem)
BOOL SearchInAllIncaDbFolders	If enabled, the MCDbProjects list in the MCSystem object is created by searching the whole INCA Db, not only the subfolders of the folder where the active workspace resides.
BOOL SearchOnlyCurrentWorkspace	If enabled, the MCDbLocations list and hence the MCDbLogicalLinks list in the MCDbProject object is creaed by using only the loaded projects in the selected workspace.
BOOL SkipWriteProtectedDatasets	If enabled, only enumerates read/write datasets. Write protected datasets are skipped. This ensures that only writable datasets are configured in INCA.
IncaLogMCD3MC LoggingMode	This option is obsolete for MCD3 MC and no longer supported with the new logging implementation in INCA 7.2. This option is still valid in ASAP3 mode.
BOOL LogExceptions	If logging is enabled, exceptions are logged (HRESULT other than S_OK)
BOOL LogCalWrite	If logging is enabled, calls to characteristic write methods are logged along with the written data.
BOOL LogCalRead	If logging is enabled, calls to characteristic read methods are logged along with the written data.

Property Type and Name	Description
BOOL LogClientCommands	If logging is enabled, all client commands are logged. Exceptions: calls to core interfaces, like AddRef/Release/QueryInterface and calls to the methods of the ___NewEnum interface
BOOL LogServerResponses	If logging is enabled, the results of all client commands are logged. Exceptions: calls to core interfaces, like AddRef/Release/QueryInterface and calls to the methods of the ___NewEnum interface
IncaLogMCD3MCInterfaceLogging InterfaceLoggingMode	Defines how logging is performed: eLogExceptions = 0xA001: Do not log at all eLogExceptionsClient = 0xA002: Log to screen only eLogExceptionsClientServer = 0xA003: Log to screen and file
BOOL LogActivityToFile	Defines if the Activity logging shall be written into the log file on disk.
BOOL LogActivityToScreen	Defines if the Activity logging shall be displayed in the ASAM MCD3 MC Window
BOOL LogInterfaceToFile	Defines if the Interface logging shall be written into the log file on disk
BOOL LogInterfaceToScreen	Defines if the Interface logging shall be written in the ASAM MCD3 MC Window
enum tagMCSTORAGETYPE MCDStorageType	This is the default value for the MCDStorageType in a newly created collector.
int MCDCollectorNumberOfSamples	This is the default value for the Number of samples in a newly created collector.
BSTR MCDCollectorStorageFile	This is the default File name that is used in case the StorageType is MCDStorageFile.
int MCDCollectorStartDelay	This is the default start delay of a newly created collector.
enum tagMCTIMERESOLUTION MCDCollectorStartDelayUnit	This is the default unit for the start delay of a newly created collector
int MCDCollectorStopDelay	This is the default stop delay for a newly created collector.
enum tagMCTIMERESOLUTION MCDCollectorStopDelayUnit	This is the default timer resolution for a newly created collector.
BOOL MCDCollectorRealTimeData	This option enables the client to receive data even if the storage type of a collector is MCDStorageAusy.
int MCDBufferDownSampling	This is the default downsampling value for the buffer of a newly created collector.
int MCDBufferRate	This is the default rate for the buffer of a newly created collector.
enum tagMCCODESCALINGUNIT MCDBufferRateUnit	This is the default rate unit for the buffer of a newly created collector.
int MCDBufferSize	This is the default size of the buffer of a newly created collector.
BOOL MCDBufferTimeStamping	This is the default value for the TimeStamping value of the buffer of a newly created collector.

Property Type and Name	Description
BOOL ShowRecordedValues	If enabled, the INCA experiment does display the recorded measurement values that are configured via ASAM MCD 3MC 4 INCA.
BOOL LoadDifferencesAfterSwitchingIgnitionOffOn	If enabled, INCA automatically downloads the differences to the ECU in case the ignition was switched off and back on again. Note that this feature is only working for specific ECUs that have no persistent storage for their data.
BOOL MaxLogFileSize	This is the maximum log file size that is generated. If this size is reached, the log file is truncated, the first 10 Kb are copied from the old file to the new one.
BOOL CreateNewDSDuringDownload	If enabled, INCA creates a new data set during the call to MCDLogicalLink::ConnectToModule(eLT_DOWNLOAD
BOOL ShowEditedValues	If enabled, the INCA experiment does display the characteristic labels that are configured via ASAM MCD 3MC 4 INCA.
BOOL ShowMeasuredValues	If enabled, the INCA experiment does display the measurement values that are configured via ASAM MCD 3MC 4 INCA.
BOOL UseHardBounds	If enabled, the hard bounds are used for validity checks. If disabled, the weak bounds are used for validity checks.
BOOL UseOneDimArray	If enabled, the read methods of MCMapCharacteristic deliver a one-dimensional array of one dimensional arays for the matrice part of the map. If disabled, a two-dimensional array is returned for the matrice.
BOOL LimitVariablesPerWindow	If enabled, the maximum nuber of measurement values per experiment measurement window is limited.
int VariablesPerWindow	If *LimitVariablesPerWindow* is enabled, this property defines the maximum number of measurement values per experiment measurement window.
int RecorderFileFormat	This property defines the format that is used in case *UseRecorderFileFormat* is enabled.
BOOL UseRecorderFileFormat	If enabled and a collector uses MCStorageFile, the results are stored in an additional file with the format given in *RecorderFileFormat*.
AlwaysUseFuliDevices	If enabled, the logical links, which are FULI-enabled and would normally appear in two instances in the DbLogicalLinks list – one of them having "#FULI" appended to the name – now only appear once in the DbLogicalLinks list but are behaving as FULI device. This is to allow for easy migration to FULI setups.

Tab. 4-1   Settings: Property Type and Name

5.3 Object Lifetime

Objects in the ASAM MCD-3MC server live as long as a client holds a reference to one of the objects of the server or any of the child objects in the servers' object tree. For example, if a client holds a reference to a *MCCollectedObject*, the parent *MCCollector, MCCollectors, MCLogicalLink, MCLogicalLinks, MCProject* and *MCSystem* objects are still alive, even if they do not have a reference held by the client. Even if an object is removed from a list (e.g., *MCCollectors.RemoveByIndex()*), it is not yet destroyed as long as a client holds a reference to it.

5.4 Client Specifics

Cache Manager: Release Objects Soon

All Objects below MCResults are created on the client side and therefore do not need any network roundtrip when being accessed. This makes the access very fast.

If the client does not release this object, the memory consumption on the client will steadily grow and finally the system will run out of memory. On Systems with a garbage collector, it is absolutely recommended to enforce the release of the MCResult trees that are not needed anymore. If this isn't done, the System will steadily run into overload situations when the Garbage collector becomes active from time to time.

With a Microsoft .net based language like C# or VB.net it is recommended to immediatelly call "System.Runtime.InteropServices.Marshal.ReleaseComObject(ObjectToRelease)" to release objects as soon as possible.

5.5 Logging

Debug logging shows class names and Interface names intermixed.

Debug logging does not show the core COM API calls, e.g., AddRef/Release/QueryInterface or the ___NewEnum calls, because they would flood the log window with entries that hold only very low information.

Logging is available in two detail levels: Activity logging and interface logging. While interface logging (also called debug logging) is useful for developers and support engineers that want to track down a certain behaviour of the system, activity logging serves as a indicator in which phase the ASAM MCD 3MC communication currently is. Activity logging shows only high-level activities, like "Project Selected" or "INCA Measurement started". Logging can be configured via the option dialog, the toolbar buttons, or the View menu.

The log file is now separated in 5 chained log files: If a fifth of the file size of the specified maximum log file size is reached for the first log file, the first log file is closed, and logging continues in the second log file. This continues until the fifth log file, after which the first log file is then overwritten. This way, at least 4/5th of the log file content is available at any time before it gets overwritten. The log files now are named following this scheme:

ASAP3_FCL*YYYYMMDDhhmmssiiii*.log

Where *YYYY* is the year, *MM* the month, *DD* the day, *hh* the hour, *mm* the minute, *ss* the second and *iiii* the millisecond of the point in time when the log file was originally created. This timestamp in the name also allows to easily determine the order of logfiles in the chain.

5.6 Events to the Client

Events are normally sent synchronously, that is, the server suspends its thread of execution until the client returns from the event. The only exceptions are the OnCollectorResultReady event and the OnCollectorError event, which are sent asynchronously in a separate server thread. Error events have higher priority and are immediately sent to the client whereas all Collector result events are stored in a queue and sent in sequential order.

5.7 Known Problems

In INCA, it is not possible to have the same description file (A2L) and data set assigned to two different ECUs at the same time. If the currently used INCA Workspace, which corresponds to the MCD3 Project, contains more than one devices, and two or more of those devices are "compatible", e.g. can have the same INCA project and data set assigned, the attempt to assign the same INCA project (=IMCLocation) and data set (=IMCDbBinary) to a device (by creating a new IMCLogicalLink) while those are already assigned to another device within INCA, will fail.

Workaround: Select a different project and/or data set to the second device – either via the MCD3-Client or via the INCA GUI - even if this device is not further used in the current session.

6 Appendix: ASAM-MCD-3MC Error Codes

The basic MCD3 error codes are described in the standard itself.

This chapter provides additional technical information about the meaning of ETAS MCD-3MC error codes.

Error code (decimal)	Description
26500	**During the execution of the method '%1' the following exception occured:\n%2** This error indicates that an exception, that is not specifically handled, was caught. Typically, this indicates problems of subsystems or internal server problems. Details can be found in the parameters %1 and %2. %1: Name of method. %2: Exception.
26501	**The Value of '%1' must be '%2' and '%3'.** This error message is shown in case that the value of a parameter is outside the allowed range. To avoid the error set the parameter to an allowed value. %1: The name of the parameter containing the value out of range. %2: The start of the allowed value range. %3: The end of the allowed value range.
26502	**Interface is not supported: '%1'.** This error message is shown in case you try to use an interface that is not supported. Typically, this indicates problems of subsystems or an internal sever problem. %1: The name of the interface that is not supported.
26503	**Member not initialized: '%1'.** This message is shown in the case that a member variable was not correctly initialized. %1: The name of the member that was not correctly initialized.
26504	**Invalid Parameter: '%1'** A parameter is invalid. %1: The name of the method the invalid parameter was assigned to or the name of the invalid parameter.
26505	**Creation of object failed: '%1'.** The creation of an object failed. %1: The name of the object that could not be created.
26506	**The index is out of range.** This message is shown if you try to access an item with an illegal index.
26507	**The return value '%1' is not initialized.** This error is shown if the return value handed over to a method call was not initialized. To avoid the error, initialize the return parameter. %1: The name of the parameter.
26508	**During the execution of the method '%1' an exception of unknown type has occured.** This error indicates that an exception of unknown type, that is not specifically handled, was caught. Typically, this indicates problems of subsystems or internal server problems. %1: Name of the method.
26509	**The parameter with the address for the return value ([retval] in IDL) contains 0 (NULL) and is therefore invalid.** This error indicates an internal server problem.

Error code (decimal)	Description
26510	**Creation of object '%1' failed. Reason code (hr): %2** This error message is shown in case the creation of an object failed. %1: The name of the object that could not be created. %2: The reason why the object could not be created.
26511	**The given Index is either less than zero or greater than the number of items in the collection minus one. The index is zero-based.** This error idicates that either the client or the server itself accessed a collection with an index value that does not point to an entry of the collection.
26512	**The collection contains an item with the value 'NULL'.** This message is shown in case an 'NULL'-item is found during iteration through a collection.
26513	**INCA returned an object of the wrong type. Expected type: '%1'** A wrong object type was returned by INCA. This indicates an interop-problem between server and INCA. %1: The type of object that was expected.
26514	**The requested functionality is not supported by FULI/MCE. Reported error by INCA:** This error occurs when the client attempts to create a characteristic object on a FULI logical link, but INCA does not allow to enable FULI functionality on that characteristic. The INCA error code is given with the message.
27102	**Storage not configured.** This error message is shown when the Check method is called, and the server encounters a missing storage object. This indicates an internal server error.
27103	**Wrong storage type.** This error message is shown when the ConfigureStorage method is called, and the server encounters an invalid storage type. This indicates an internal server error.
27104	**Operation is not allowed in state(s): '%1'.** This error message is shown in case the collector is not in the correct state (neither activated nor started), e.g. in the call to CreateClientIDForPolling(). %1: The ID of the corresponding collector.
27105	**Operation requires sta te(s): '%1'.** This error message is shown in case the object is not in the correct state for the used operation. %1: The state that is required for the operation.
27106	**Operation is not allowed while other Collectors or Recorders are in state 'eOS_ACTIVATED' or 'eOS_STARTED'.**
27200	**The MCDBuffer default rate of %1 selected in the 'Options' dialog is not available through the current MCDLogicalLink. The nearest raster %2 was selected instead.** Ths message indicates that a collector object is added by the client and the server can not match the default rate from the 'Options' dialog with one of the current MCDLogicalLink MCDRasters list. In that case, the "nearest" raster is used, eg. instead of the default 100 ms raster, a 120ms Raster is chosen. %1: The default rate set within the 'Options' dialog %2: The selected rate
27301	**IncaObjectManager failed to add INCA variable '%1'. Raster table '%2' seems to be full.** This error indicates that INCA could not wire a requested variable. Typically, this indicates that the chosen raster table for the ECU is already full and has no free

Error code (decimal)	Description
	entries for the variable available. To measure the variable, another collector with a different raster %1: Name of the variable. %2: Name of the raster table.
27304	**IncaObjectManager failed to add variable '%1' with rate '%2', because it is already wired in raster '%3'. Virtual variables only work between equidistant rates!** This message indicates that a limitation was detected which does not allow to wire this variable as a downsampled dependent value. %1: Name of the variable. %2 The rate.
27305	**IncaObjectManager failed to add variable '%1' with rate '%2'. Virtual variables are not possible in CAN-Monitoring or CalcDev or FULI devices!** This message indicates that a limitation was detected which does not allow to wire this variable as a downsampled dependent value. %1: Name of the variable. %2 The rate.
27306	**IncaObjectManager failed to add variable '%1' with rate '%2', Downsampling of Recorder variables is not possible!** This message indicates that a limitation was detected which does not allow to wire this variable as a downsampled dependent value. %1: The name of the variable. %2: The rate.
27307	**IncaObjectManager failed to add variable '%1' with rate '%2', Downsampling not possible because rewiring of Recorder or FULI/MCE variables not allowed!** This message indicates that a limitation was detected which does not allow to wire this variable as a downsampled dependent value. %1: Name of the variable. %2: The rate.
27308	**INCA had the following error during Measurement/Recording: '%1'** This is an asynch error that is logged if the object manager detects inconsistencies with the TargetServer interface. Examples are: – - Timestamps differ more than 50% from the raster size; - older timestamp detected (= timestamps monotony violated); – - Timestamp dump forced by previous errors; – - Raster %s had no new data for more than %ld ms; - Amount of samples for virtual wired variable %s varies from the measured amount of samples for the rate %s by more then one sample; ... %1: The error.
27309	**IncaObjectManager failed to add variable '%1' with Representationtype '%2', because it is already wired with a different Representationtype!** The variable is already wired in another collector but with a different rate and a different representation type. Virtualisation of channels is only possible between variables with identical representation type. %1: Name of the variable. %2: The representation type.
27310	**During the 'Check' of the Collector the following problems occured:** This message is followed by the list of all errors that occurred during the execution of the 'Check' command. Typically, these are wiring errors like full rasters or incompatible variables for virtualization. The message is logged and is returned within the error object parameter of the Check() method.

Error code (decimal)	Description
27311	**IncaObjectManager failed to add the following INCA variable(s) '%1', because of full Raster tables.** This message indicates that the raster table for this collector is full and cannot hold the given variables for measurement. %1: The list with the names of the variables.
27312	**IncaObjectManager failed to add variable '%1' with rate '%2', because it is already wired in raster '%3'. Virtual variables only work between equidistant rates!** This message indicates that the variable was already wired in a different raster but either that other raster or the current requested raster is not an equidistant one. %1: Name of the variable. %2: The requested rate. %3: Name of the other raster.
27313	**IncaObjectManager failed to add the variable(s) '%1', because virtual variables are not possible in CAN-Monitoring or CalcDev or FULI devices!** This message indicates that the variable cannot be wired as virutal with the current logical link because virtual variables are not supported with that device type. %1: The name of the variable.
27314	**IncaObjectManager failed to add the variable(s) '%1', because downsampling to variables with a Collectors storage type eST_FILE is not possible!** This message indicates a limitation which prevents wiring of an already wired variable in a different raster. %1: The name of the variable that could not be added.
27315	**IncaObjectManager failed to add the variable(s) '%1', because downsampling not possible because rewiring of variables with a Collectors storage type eST_FILE not allowed!** This message indicates a limitation which prevents wiring of an already wired variable in a different raster. %1: The name of the variable
27316	**IncaObjectManager failed to add the variable(s) '%1' because it is already wired with a different Representationtype!** This message indicates a limitation which prevents wiring of an already wired variable in a different raster. %1: The name of the variable.
27317	**The Buffer is too small for the selected Startdelay.** The number of samples given for the buffer size is too small to cover all data that is required to be held for the requested Startdelay. Please either increase the buffer size or reduce the Startdelay.
27319	**IncaObjectManager failed to add variable '%1', because INCA returned ErrorCode 515 when setting the variable display state!** %1: Name of variable (label). This indicates an INCA error.
27320	**Variable %1 could not be downsampled, as the reference raster is full.** %1: Name of variable (label). The error indicates that the variable cannot be wired because there is not enough available space in the requested raster, and not in a faster raster that otherwise might be used as a downsampling base.
27500	**ObjectManager failed to retrieve the Global Interface Table pointer of the Collector!** This message in the log file indicates an internal error of the server.
28000	**The calibration object can't be selected in INCA.**

Error code (decimal)	Description
	This message indicates that INCA was not able to select the specified calibration object.
28001	**The selected object is not a scalar.** This message indicates that INCA reports that the given characteristic is not a scalar characteristic. Internal server error or INCA error.
28002	**Can't get HEX values from INCA because of unknown data type.** This message indicates that the calibration object has an unsupported data type and hence no data from INCA can be read.
28003	**It is not possible to write HEX values to INCA because of unknown data type.** This message indicates that the calibration object has an unsupported data type and hence no data can be written to INCA.
28004	**One of the values which have to be set in the MC system is out of the limits of the data type. The command can't be executed!** This message indicates that during a request to write calibration data, it has been detected that the data to write exceeds the limits of the data type of the calibration object and hence the data cannot be written.
28005	**One of the values can't be set in the MC system because of an invalid data type!** This message indicates that a write method was called with a VARIANT data type that is not supported. Valid data types are VT_I4, VT_INT, VT_UI4, VT_UINT, VT_BOOL, VT_R4 and VT_R8.
28006	**One or more values from the MC system can't be interpreted because they are invalid!** This message indicates that INCA delivered a calibration object with a structure that does not match the expected structure of the MCD3 server. This includes different dimensions, empty values, error values, scalar instead of array types and NULL values.
28007	**Because of write protection the scalar value can't be changed!** This message indicates that a scalar characteristic is write protected and cannot be written.
28008	**Because of write protection the vector values can't be changed!** This message indicates that a vector characteristic (e.g., curve or val_blk) is write protected and cannot be written.
28009	**The index which is used to have access to the data is out of the possible limits!** This message indicates that a write access to a matrix, vector or value block was attempted with an index value that is out of bounds of the object.
28010	**The StopIndex must be bigger than the StartIndex!** This message indicates that an attempt was made to perform a read or write access to a matrix, vector or value block object with an invalid StartIndex and StopIndex combination: StopIndex must always be larger or equal to StartIndex!
28011	**It is not possible to change rescale objects with ASAM MCD 3MC!** This message indicates that write access to rescale characteristic objects is not possible.
28012	**Because of write protection the matrix values can't be changed!** This message indicates that a write access to a matrix calibration object was performed, which is write protected. The write access failed.
28013	**During the execution of the method '%1' the following error occurred: %2** This is a generic error text which will be filled with concrete errors and a method name in which the error occurred. %1: The name of the method %2: The error that occurred

Error code (decimal)	Description
28014	**The selected object is not a curve!** This message indicates an inconsistency: A presumably curve characteristic reports that it is not a curve. The object cannot be created.
28015	**The curve object is not available!** This message indicates an inconsistency in the server: The value object for a curve object is not initialized.
28016	**The axis object is not available!** This message indicates an inconsistency in the server: The value object for an axis object is not initialized.
28017	**The map object is not available!** This message indicates an inconsistency in the server: The value object for a map object is not initialized.
28018	**Because of write protection the curve value(s) can't be changed!** This message indicates that write access to write protected curve values is not possible.
28019	**Because of write protection the axis value(s) can't be changed!** This message indicates that write access to write protected axis values is not possible.
28020	**Because of write protection the map value(s) can't be changed!** This message indicates that write access to write protected map values is not possible.
28021	**The selected object is not a map!** This message indicates an inconsistency, because a presumably map object reports that it is not a map object.
28022	**Object has been disconnected from the parent!** This message indicates an inconsistency in the server, because the accessed characteristic object does not have the mandatory parent object set.
28023	**DB object not available!** This message indicates that an attempt was made to add a characteristic to the Characteristics collection, but the provided object was no DbCharacteristic (nor derived) object. The Add method requires the first parameter to be a valid DbCharacteristic object to succeed.
28024	**Can't get object ""LogicalLinks"** This message indicates an inconsistency in the server, because the server was not able to access the LogicalLinks collection via the parent chain of the characteristic.
28025	**The calibration object was already selected! Please use the method ""GetItemByName"" or GetItemByIndex"" to obtain access to the object!** This message indicates that an attempt was made to add a characteristic to the Characteristics collection which is already contained in the collection. To retrieve the contained object, use one of the methods GetItemByName or GetItemByIndex instead.
28026	**The scalar characteristic can't be created!** This message indicates that the creation of the characteristic object during the "Add" method call failed. The log file may contain additional info about the failure.
28027	**Characteristics Interface not supported!** This message indicates an inconsistency in the server because a characteristic object does not provide the corresponding interface.
28028	**INCA calibration object can't be created!** This message indicates an internal error: The calibration object could not be created. The log file may contain additional info.

Error code (decimal)	Description
28029	**The curve characteristic can't be created!** The message indicates an internal error: The curve calibration object could not be created. The log file may contain additional info.
28030	**The map characteristic can't be created!** The message indicates an internal error: The map calibration object could not be created. The log file may contain additional info.
28031	**The selected characteristic type will not be supported (only scalars, curves and maps can be selected)!** The message indicates that either an attempt was made to create a runtime characteristc from a DbCharacteristic which is not supported or that INCA indicates an unsupported characteristic type during the creation DbCharacteristic objects.
28033	**Can't get locked Experiment View from INCA!** This message indicates an internal error in the server or in INCA.
28034	**Can't attach Experiment from INCA!** This message indicates that the server was not able to get the INCA experiment from the experiment view. This is an internal error of the server and/or INCA.
28035	**Can't get Experiment Manager from INCA!** This message indicates an inconsistency in the server.
28036	**Can't get object "ProjectInternal"!** This message indicates an internal server problem.
28037	**Can't get object "LogicalLinks"!** This message indicates an internal server problem.
28038	**Can't get object "Characteristics"!** This message indicates an internal server problem.
28039	**The short name or the long name of the calibration object can't be extracted from the data base object!** This message indicates an internal server problem.
28040	**It's not possible to write values because the Reference Page is active. Please switch first to the Working Page!** This message indicates that an attempt to write characteristic values was made but the active page is the reference page. Only the working page allows for write access.
28041	**Can't get a valid workbase device!** This message indicates an internal server problem.
28046	**It is not possible to change CURVE_AXIS objects with ASAM MCD 3MC!** This message indicates that it is not possible to write to curve axis characteristic objects.
29000	**The given first parameter was no DbLogicalLink!** This message indicates that the method AddByObjects or AddByObject at the LogicalLinks collection was called with an object that is no DbLogicalLink object.
29001	**%1 not found in database.** The message is shown if an element (e.g.: "DbVehicleInformation") was not found in the database. %1: The name of the element.
29002	**%1 not available.** The error message can be shown in case one of the following objects is not available during handling logical links (DbBinaries, DbLocation, DbProject, DbBinaries, DbLocation) and indicates an internal server inconsistency.
29003	**Parent is not a MCDProject.** This message indicates an internal error of the server.

Error code (decimal)	Description
29004	**Parent is NULL (not set).** The error message is shown in case the parent of the object is 'NULL'. It indicates an internal error of the server.
29005	**Unable to get Experiment from IncaExperimentView!** This message indicates that it was not possible to retrieve the Experiment from the IncaExperimentView object. This is either an internal server error or an INCA error.
29008	**Unable to set Project and DataSet in INCA.** This message indicates an internal server error or an INCA error.
29009	**Unable to get data set '%1' from INCA.** This message indicates an internal server error. %1: The name of the data set.
29010	**Unable to set INCA project in INCA.** This error message is shown in case an INCA project could not be set for a workbase device. It indicates an internal server error or INCA error.
29011	**Cannot create or initialize MCDRateInfo object '%1'!** %1: The name of the MCDRateInfo object.
29012	**Invalid element type in array returned by '%1'. Expected type: '%2'** This error is shown in case an element with an invalid type was detected within an array. It indicates an INCA error. %1: The name of the method the error occurred in %2: The type of that was expected for the element
29013	**INCA returned wrong array dimensions for '%1'.** This error message is shown in case the number of elements found in an array differs from the expected one. It indicates an internal server error or an INCA error. %1: The name of the method the error occurred in
29014	**INCA returned wrong data type (no SAFEARRAY) for '%1'.** This error message indicates an INCA error. %1: The name of the method the error occurred in
29015	**Unable to get ExperimentDevice object from INCA.** This message indicates either an internal server error or an INCA error, because INCA did not find a device with the given name in the Experiment.
29016	**Unable to get ShortName of MCDbPhysicalVehicleLinkOrInteface object.** This message indicates an internal server error.
29017	**Unable to get MCDbPhysicalVehicleLinkOrInterface object** This message indicates an internal server error.
29018	**Unable to access IncaExperimentMgr!** This message indicates an internal server error.
29019	**Unable to access IncaExperiment** This message indicates an internal server error.
29022	**Unable to upload working page!** This error message is show in case the upload of the working page failed (and no specific error report from INCA was available).
29023	**Unable to connect to module because memory page contents are different in PC and target.** This message indicates that the ConnectToModule method was called with the parameter "Type" set to "eLT_ERROR" and differences between the Target emulation memory and the INCA data set (DbBinary) have been detected by INCA. The logical link cannot connect to the Target in this case.
29024	**Unknown MCLOADINGTYPE value!**

Error code (decimal)	Description
	This error message is shown in case an invalid MCLOADINGTYPE was provided in the method call ConnectToModule.
29025	**Unable to create hex file for work page and code!** This error message is shown in case the method "SaveBinaryToFile" fails (and no specific error report of the workbase device is available from INCA).
29026	**This logical link contains no workbase device, only experiment device.** This message indicates that an attempt was made to change the MemoryPage property to a value > 1 at a LogicalLink object that does not support different memory pages.
29027	**Workbase device requested but IncaExperimentMgr not locked!** This message indicates an internal server error.
29028	**Parameter '%1' has the invalid value NULL (Nothing)!** This message indicates that a parameter that shall have a certain non-null value was not provided with the value null. %1: The name of the parameter containing the invalid value
29029	**The type or SAFEARRAY structure for initializing the Value property of the IMCRateInfo object is wrong.** This message indicates an internal server error.
29030	**Unable to create vector in SafeArray for Value property. Out of memory?** This message indicates either an internal server error or an out of memory situation.
29031	**INCA returned non-dataset item in dataset list!** This message indicates an error in INCA which occured during the creation of the list of available DbBinaries.
29032	**INCA returned no object or wrong object type for active Hardware Configuration. Please mark a Hardware Configuration in your database as active! The search for INCA Workspaces will be performed now in the whole INCA database.** This message indicates that there was no Hardware Configuration marked active within INCA. Therefore, the search of INCA workspaces cannot be limited to the folder of the active workspace and hence the whole database will be searched.
29033	**No MCDDbVehicleInformation object available!** This message indicates an internal server error.
29034	**Unable to access first MCDDbVehcleInformation object!** This message indicates an internal server error.
29035	**The given object is not an MCDDbVehicleInformation from the MCDDbProject this MCDProject was created from.** This message indicates that in the call to SelectDbVehicleInformation, a DbVehicleInformation object was provided that does not belong to the current MCSystem object.
29036	**Unable to get short name of MCDDbVehicleInformation object!** This message indicates an internal server error.
29037	**Unable to find MCDDbVehicleInformation named '%1'!** This message indicates that the name for the MCDDbVehicleInformation was not found and therefore is invalid. %1: The name of vehicle information
29038	**Connection to client revoked, unable to start ASAM MCD 3MC session while INCA is set to ASAP3 V2.1 mode. Please switch INCA to ASAM MCD 3MC mode first (User Options/General).**

Error code (decimal)	Description
	This message indicates that INCA is set to ASAP3 V2.1 mode while a client tried to start and connect to the MCD3 V1.0.1 server. INCA needs to be switched to the correct protocol first.
29039	**No project to deselect!** This message indicates that a call to DeselectProject has been made while no Project object is available. This might indicate either an internal server error or a call to DeselectProject in the wrong state of MCSystem.
29040	**No INCA connection available!** This error message indicates an internal error of the server.
29041	**ConnectCounter in MCDSystem is not 0 in ePROJECT_SELECTED state!** This error message indicates an internal error of the server.
29042	**MCDSystem object not in eVIT_SELECTED state but an attempt is made to connect a MCDDbLogicalLink!** This error message indicates an internal error of the server.
29043	**MCDSystem object not in eLOGICALLY_CONNECTED state but an attempt is made to disconnect a MCDDbLogicalLink!** This error message indicates an internal error of the server.
29044	**The ConnectCounter in the MCDSystem object is below or equal 0, but an attempt is made to disconnect a MCDDbLogicalLink!** This error message indicates an internal error of the server.
29045	**INCA reports too less elements for Asap2Characteristics_Dispatch::GetAxisDescriptions.** This message indicates either an internal server error or an INCA error.
29046	**Unable to access the Asap2Project object '%1' of the INCA database!** %1: The name of the Asap2 project. This message indicates that INCA cannot get the requested ASAP2 project from the INCA database. This either indicates an internal server error or an INCA error.
29047	**Unable to get access to the current INCA database! Is there another application using INCA remotely?** This message indicates either an INCA error or the fact that a second application uses the INCACOM API.
29049	**Unable to access the Asap2Module object of the INCA database!** This message indicates either an internal server error or an INCA error.
29050	**INCA returned a measurement object that has no name!** This message indicates an INCA error.
29051	**INCA returned a list of object references that contains a NULL reference when calling '%1'** %1: The name of the method that was called. This message indicates an INCA error.
29052	**INCA returned a characteristic object that has no name!** This message indicates an INCA error.
29053	**Unable to initialize charactieristic data of object '%1'.** This message indicates that a server internal error occurred during initialization of a characteristic object. %1: The name of the object.
29054	**Unable to initialize axis data of characteristic object '%1'.** This message indicates that a server internal error occurred during initialization of a characteristic object. %1: The name of the object
29056	**Unable to get item with index '%1' from collection '%2'. Code: %3**

Error code (decimal)	Description
	This message indicates that the server encountered an internal error during the access to an item from a collection. %1: The index of the item %2: The name of the collection %3: The error code
29057	**Unable to get ShortName or LongName from MCDDbPhysicalVehicleOrInterface object '%1'!** This message indicates an internal server error. %1: The name of the object
29058	**Unable to initialize object '%1'.** This message indicates an error during initalization of the MCDDbProject or MCDDbProjectDescription object. %1: The name of the object.
29059	**Unable to add MCDDbPhysicalVehicleLinkOrInterface object to MCDDbPhysicalVehicleLinkOrInterfaces list below the MCDDbVehicleInformation object! Reason code: %1** This message indicates that an internal error occurred when filling the list of available interfaces. %1: The reason code
29061	**INCA returned no object or wrong object type for folder: '%1'. Failed to create MCDDbProject object!** This message indicates an INCA error. %1: The name of the folder
29062	**INCA is unable to return a Hardware Configuration object with the name '%1' of the selected project! Is INCA database corrupt?** This message indicates that an error occurred while retrieving the HardwareConfiguration object from INCA. %1: The name of hardware configuration object.
29063	**Unable to get the INCA Experiment view object!** This message indicates a server internal error.
29064	**Unable to get the INCA Online Experiment from its view!** This message indicates that the server was not able to retrieve the IncaOnlineExperiment object from the INCA API.
29066	**Unable to close INCA experiment view!** This message indicates that INCA reported an error when the server tried to close the INCA Online Experiment.
29067	**Unable to get parent folder of INCA hardware configuration!** This message indicates that the server was not able to retrieve the parent folder of an HardwareConfiguration object from the INCA API.
29068	**Unable to create INCA experiment environment. Database corrupt?** This message indicates an INCA error.
29069	**Unable to set INCA experiment environment to hardware configuration. Database corrupt?** This message indicates an INCA error.
29070	**Unable to get the INCA experiment environment that is assigned to the hardware configuration!** This message indicates an INCA error.
29071	**Unable to set INCA hardware configuration to experiment environment!** This message indicates an INCA error.
29072	**Unable to save currently open INCA experiment!**

Error code (decimal)	Description
	This message indicates that INCA does not allow to save the currently open experiment.
29073	**Unable to open INCA experiment view** This message indicates that INCA was not able to open the experiment.
29074	**Unable to stop INCA recording!** This message indicates that it was not possible to successfully send a manual stop trigger to INCA.
29075	**Unable to stop INCA measurement!** This message indicates that INCA fails when the server requests to stop the measurement.
29076	**Unable to switch INCA calibration access off! For more information, plesae consult the INCA Monitor window / log file.** This message indicates that INCA has a problem when switching the INCA calibration access off. Details of this error can be found in the INCA Monitor log.
29077	**Unable to add logical link '%1' to MCDDbLogicalLinks!** This message indicates an error during the creation of a new logical link. Depending on the type of error, previous entries in the server log contain further details. %1: The name of the logical link
29078	**Unable to deliver a link to the INCA experiment view because the view is temporarily closed for logical link creation.** This message indicates an internal error of the server.
29079	**LogicalLink pointer not registered!** This message indicates an internal error of the server.
29080	**Too many GlobalObjectDeactivated() calls!** This message indicates an internal error of the server.
29081	**Unable to retrieve IncaOnlineExperiment!** This message indicates an INCA error.
29082	**Unable to put new online state while experiment is closed for link creation!** This message indicates an internal server error.
29083	**Logical Link must be in state eOFFLINE to be able to switch it online (connect)!** This message indicates that the logical link is in the wrong state for the current operation. The link has to be eOFFLINE but was found to be either eONLINE_RUNNING or eONLINE_IDLE.
29084	**Unable to switch calibration access on!** This message indicates that INCA was not able to switch the calibration access on. Please consult the INCA Monitoring window/log for more details.
29085	**Unable to switch INCA calibration access on because recorder is running!** This message indicates that INCA was not able to switch the calibration access on because there is a recorder running.
29086	**Logical Link must be in state eONLINE_IDLE to be able to switch it offline (disconnect)!** This message indicates that the LogicalLink must be in state eONLINE_IDLE so that it can be disconnected. The logical link is in state eONLINE_RUNNING or eOFFLINE.
29087	**Unable to switch calibration access off!** This message indicates that INCA was not able to switch the calibration access off. Please consult the INCA Monitoring window/log for more details.
29088	**Empty Data Set name found in INCA!** This message indicates an INCA error.
29089	**Unable to get Data Set with name '%1'.**

Error code (decimal)	Description
	This message indicates either an internal server error or an INCA error. %1: The name of the data set
29090	**Failed to load MCDDbLocation object with data from INCA. Reason: %1** This message indicates that the server had a problem while loading the ASAP2 information from INCA. This can be either because of an internal server error or an INCA error. %1: The reason code
29091	**Failed to initialize MCDDbBinaries with data from INCA. Reason: %1** This message indicates that the server failed to initialize the collection of binaries. Previous messages in the log contain further information about the error. %The reason.
29092	**No device name available from INCA!** This message indicates an INCA error.
29094	**Unable to create subfolder in INCA.** This message indicates either an internal server error or an INCA error.
29095	**Unable to create folder for workspace with given ShortName in INCA database! Check naming conventions and probably INCA database consistency.** This message indicates an error during creation of the folder for a given workspace name when adding a project. The previous entries in the server log contain further information. Probably a invalid worspace name (or path within the INCA database) was given.
29096	**Unable to create new HardwareConfiguration object in INCA database!** This indicates an INCA error.
29097	**Unable to create INCA workspace. Reason: %1** This message indicates an error during workspace creation. The previous entries in the server log and/or the INCA Monitor window/log contain further information. %1: The reason
29098	**Unable to create MCDDbProject object. Reason: %1** This message indicates an error during MCDDbProject creation. The reason code is shown. This might be either an internal server error, an INCA error, or an invalid project name. %1: The reason
29099	**Unable to add MCDDbProjectDescription object. Reason: %1** This message indicates an error during MCDDbProjectDescription creation. The reason code is shown. This might be either an internal server error, an INCA error, or an invalid project name. %1: The reason
29100	**No Active DbProject in configuration!** This message indicates that an attempt was made to get the active DbProject while in Db configuration mode, but there is no active DbProject. This indicates an internal server error.
29101	**Close only allowed if MCDSystem is in state eDBPROJECT_CONFIGURATION** This message indicates an internal server error.
29102	**MCDSystem object is in an unsupported state!** This message indicates an internal server error.
29103	**MCDSystem state is eDBPROJECT_CONFIGURATION but no DbProject set in MCDDbProjectConfiguration!** This message indicates an internal server error.
29104	**Loading or adding a MCDbProject is only allowed in MCDSystem mode eINITIALIZED or eDBPROJECT_CONFIGURATION.**

| Error code (decimal)	Description
	This message indicates that the client attempted to enter the MCDDbProjectConfiguration while in one of the states ePROJECT_SELECTED or eLOGICALLY_CONNECTED. The clients need to deselect the current project first.
29105	**INCA folder '%1' does not exist!** This indicates an error in INCA. %1: The name of the folder that does not exist
29106	**INCA hardware configuration '%1' does not exist!** This message indicates either an internal server error or an error in INCA. %1: The name of the hardware configuration that does not exist
29107	**Unable to delete hardware configuration '%1' in INCA database!** This message indicates an error in INCA: %1: The name of the hardware configuration
29108	**Unable to remove hardware configuration '%1' from INCA database!** This message indicates that the client gave a wrong index for the remove method. See the server log for further details. %1: The name of the hardware configuration
29109	**Unable to create new ASAP2 project in INCA database!** This message indicates an error while reading the ASAP2 project file from disk into the INCA data base. Please consult the inca Monitor window or log for further details.
29110	**Add only allowed if MCDSystem is in state eDBPROJECT_CONFIGURATION** This message indicates that the client tried to add a DbLocation or DbBinary element, but the system state is not the required eDBPROJECT_CONFIGURATION.
29111	**Unable to rename the INCA Asap2 project to the given ShortName '%1'!** This message indicates that an error occurred while the client tried to create a new ASAP2 project. See INCA Monitor window/log for further details. %1: The not set short name of the project
29112	**Unable to create ASAP2 project folder for data set within INCA database!** This message indicates that an error occurred while the server was trying to add a ASAP2 project folder. Either the folder name is invalid, or the server has an internal error. See previous server log entries for further details.
29113	**Unable to create new data set in INCA database!** This message indicates that an error occurred while the server was trying to add a hex file. Either the data set name is invalid, or INCA has an error. See INCA Monitor window / log for further details.
29114	**Unable to find folder of ASAP2 project within INCA database!** This message indicates an error while removing an MCDDbLocation. This is either an internal server error or an INCA error.
29115	**Add of binary only allowed if MCDSystem is in state eDBPROJECT_CONFIGURATION** This message indicates that the client tried to add a MCDDbBinary while the system is not in the required state eDBPROJECT_CONFIGURATION.
29117	**Unable to delete data set '%1'!** This message indicates an error in INCA. %1: The data set that could not be deleted
29118	**An item with name '%1' does already exist in the INCA database!** This message indicates that while adding a new workspace, INCA detected that the workspace already exists within its database. %1: The name of the already existing data base object
29121	**Unable to create new MCSystem object, only one object at a time supported.**

Error code (decimal)	Description
	This message indicates that a second client connection was refused by the server since the server has already a client connection.
29122	**Unable to create Logical Link: Other Logical Links are already in state eONLINE_RUNNING and INCA does not allow for the creation of new LLs while measurement is running** This message indicates that there is already a measurement running within INCA and addition of new logical links is not possible in this state.
29124	**Unable to download Workbase (Working- and/or Reference- and/or Codepage)** This message indicates that an error occurred during the download of the Workbase to the target. Please consult INCA Monitoring window / log for futher details.
29125	**Unable to find project with name '%1' in database!** This message indicates that the server did not find the MCDDbProject item that was requested in a SelectProjectBy*X* method call. %1: The name of the project that could not be found
29126	**Unable to find item '%1' in collection!** This message indicates that an attempt was made to get or remove a certain item in a collection, but the item was not found in the collection. %1: The item that could not bet found in the collection
29127	**There is already a LogicalLink open with the same INCA device but with different FULI/MCE attribute. Both LogicalLinks must use same Binary. Other Link's Binary name: '%1'** This error indicates that a call to one of the LogicalLinks.AddByXXX() methods was performed with parameters that would result in a LogicalLink that would be identical to an already existing logical link, but with different FULI/MCE attribute and with a different Binary. This is not allowed, the Binaries must be identical in this case, since both LogicalLinks point to the same physical device. %1: The name of the other binary.
29128	**There is already a LogicalLink open with the same INCA device but with different FULI/MCE attribute. Both LogicalLinks must use same Binary. Other Link's Binary name: '%1'** This error indicates that a call to one of the LogicalLinks.AddByXXX() methods was performed with parameters that would result in a LogicalLink that would be identical to an already existing logical link, but with different FULI/MCE attribute and with a different Binary. This is not allowed, the Binaries must be identical in this case, since both LogicalLinks point to the same physical device. %1: The name of the other binary.
29129	**LogicalLink cannot be added because there is already a LogicalLink object with the same short name in the LogicalLinks collection!** This message indicates that already a logical link with the given name is in the LogicalLinks collection. A MCLogicalLink can be instantiated only once.

Tab. 5-1   Error codes

7 Contact Information

Technical Support

For details of your local sales office as well as your local technical support team and product hotlines, take a look at the website: www.etas.com/hotlines
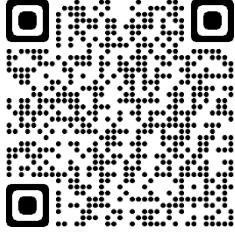
ETAS Headquarters

ETAS GmbH

Borsigstraße 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	Internet:	www.etas.com

Figures

Tables