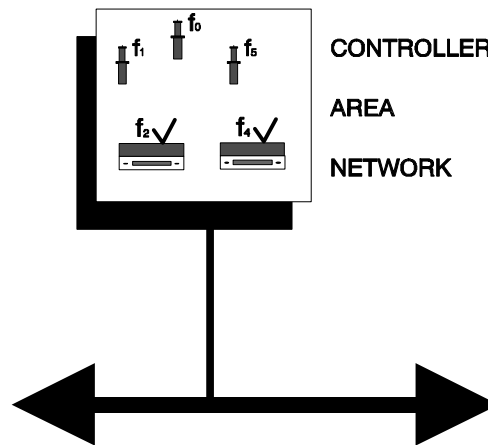


**CAN in Automation (CiA)
International Users and Manufacturers Group e.V.**



**CAN Application Layer for Industrial Applications
CiA/DS207
February 1996**

Application Layer Naming Conventions

1. SCOPE

This document contains the naming conventions that apply to instances of the objects that are defined by the service elements of the CAN Application Layer. This document is part of a set of documents that standardize the CAN Application Layer for Industrial Applications.

2. REFERENCES

/1/: CiA/DS201, CAN Reference Model

3. GENERAL DESCRIPTION

3.1 Naming Conventions Perspective

The CAN Application Layer (see /1/) offers an open CAN network where modules from different suppliers cooperate in a distributed application. To this purpose, the service elements of the CAN Application Layer (see /1/) model their functionality through the use of objects. Applications can create instances of those objects identified by a *name*. These names have a *network-wide scope*. Applications that want to execute remote services via the network on such an object must know its name and this name must identify the object.

3.2 Symbolic Name

A symbolic name is built according to the following syntax rules:

```
<symb-name>      ::= { <alpha-num> | <special> }
<alpha-num>      ::= 'A' | ... | 'Z' | 'a' | ... | 'z' | '_' | <num>
<num>            ::= '0' | ... | '9'
<special>        ::= '#'
```

If a symbolic name is transferred via the CAN network, each character is transferred in one data byte of the COB as an 8-bit unsigned integer, whose value is the ASCII code of that character. The bits in this byte are transferred most significant bit first. The character sequence is transmitted from left to right. Symbolic names are case sensitive.

3.3 Byte Selector

A byte selector is a non-negative integer that can assume the following values:

```
<byte-selector> ::= 1 | ... | 255
```

If a selector is transferred via the CAN network, it is transferred in one data byte of the COB as an 8-bit unsigned integer, whose value equals the value of the selector. The bits in this byte are transferred most significant bit first.

3.4 BCD Number

A BCD number is built according to the following syntax rules:

```
<bcd-num>          ::= { <bcd-pair> }
<bcd-pair>         ::= <bcd-digit-1><bcd-digit-0>
<bcd-digit-1>     ::= <bcd-digit>
<bcd-digit-0>     ::= <bcd-digit>
<bcd-digit>       ::= 0 | ... | 9
```

If a BCD number is transferred via the CAN network, each BCD pair is transferred in one data byte of the COB as an 8-bit unsigned integer, whose value is defined as $16 * \text{<bcd-digit-1>} + \text{<bcd-digit-0>}$. The bits in this byte are transferred most significant bit first. The sequence is transmitted from left to right. A <bcd-pair> represents the value $10 * \text{<bcd-digit-1>} + \text{<bcd-digit-0>}$.

4. CMS NAMING CONVENTIONS

4.1 Purpose

CMS is one of the service elements of the CAN Application Layer, see /1/. CMS defines a number of objects and remote services on these objects. In order to implement the remote services two (or more) peer CMS entities have to exchange information using a protocol. Such a protocol uses COB's each with a unique COB-ID, to transfer the protocol data. Such A COB-ID uniquely identifies the CMS object. All clients and servers of a CMS object must use the same COB-ID for the COB's used by the protocol of that CMS object.

In the CAN Application Layer the COB-ID's of CMS objects may be distributed by the Distributor service element (DBT, see /1/). The DBT distributes COB-ID's based on the *names of the COB's* used by the protocol of that CMS object. The CMS naming conventions serve to control the distribution of COB-ID's by the DBT.

4.2 CMS Object Name Syntax

A CMS object name is a symbolic name. The Client(s) and Server(s) of a CMS object do not necessarily use the same name for an object. The DBT service element has the possibility to 'link' these names to the same CMS object.

The name of a CMS object must adhere to the following syntax:

```

<CMS_obj_name>      ::= <fix-format-name> | <free-format-name>
<fix-format-name>   ::= <CMS-prof-ID> <appl-spec-name> <CMS-node-ID>
<CMS-prof-ID>      ::= { <alpha-num> }3
<appl-spec-name>    ::= { <alpha-num> }7
<CMS-node-ID>      ::= { <num> }3
<free-format-name> ::= '#' { <alpha-num> }12

```

All CMS object names consist of thirteen characters. All names that start with a '#' are free format names. Names that do not start with a '#' are names whose format is fixed by CiA.

The <CMS-prof-ID> is a sequence of three alpha-numeric characters. It indicates the profile-ID or application type of the CMS object. Profile-ID's are assigned and registered by the CiA. The profile-ID "000" is to be used by CMS objects that do not belong to a registered profile.

Application Layer Naming Conventions

The <appl-spec-name> is a sequence of seven alpha-numeric characters and can be chosen freely by the application unless further conventions are defined for a certain profile-ID.

The <CMS-node-ID> is a sequence of three alpha-numeric characters that represents symbolically, from left to right, the value of the Node-ID of the NMT Slave where the CMS object is used. The Node-ID of a module is defined by the NMT Service Element, see /1/. A <CMS-node-ID> equal to "000" may be used to generate CMS object names that are independent from the Node-ID of the NMT Slave where they are used.

4.3 COB Name Syntax

A COB name is a symbolic name consisting of fourteen characters. The names of the COB's that are used by a protocol of a CMS object are defined by the following rules:

- If the protocol of a CMS object requires one COB, the name of that COB is equal to the name of that CMS object concatenated with the character 'X'.
- If the protocol of a CMS object requires two COB's, the COB transmitted from Client to Server is equal to the name of that CMS object concatenated with the character 'C'.
- If the protocol of a CMS object requires two COB's, the COB transmitted from Server to Client is equal to the name of that CMS object concatenated with the character 'S'.

```

<COB-name>          ::= <X-COB-name> | <C-COB-name> | <S-COB-name>
<X-COB-name>       ::= <CMS-name> 'X'
<C-COB-name>       ::= <CMS-name> 'C'
<S-COB-name>       ::= <CMS-name> 'S'

```

4.4 CMS Data Type Names

CMS Data Type Names are symbolic names for CMS Data Types. CMS Data Type Names are not transferred via the CAN Network.

5. NMT NAMING CONVENTIONS

5.1 Purpose

The NMT naming conventions serve to identify the NMT objects that can be managed via remote services on these objects.

5.2 NMT Object Name Syntax

Since there can be only one network object in a CAN network, it does not require a name.

An NMT Slave is identified by an *NMT-Address*. An NMT-Address consists either of a *module-ID* or of a *module-name* and a *module-ID*. A module name is a symbolic name. A module-ID is a selector. They adhere to the following syntax:

```
<NMT-Address> ::= <module-name> <module-ID>|<module-ID>
<module-name> ::= { <alpha-num> }7
<module-ID>   ::= <byte-selector>
```

An NMT-Address can be configured on an NMT Slave by the LMT Service Element (see /1/) or by module specific means that do not fall within the scope of the CiA Standard on the CAN Application Layer. For an NMT-Address at least one of the following conditions must be met:

- there is no other NMT Slave in the network with the same <module-name>
- there is no other NMT Slave in the network with the same <module-ID>

6. LMT NAMING CONVENTIONS

6.1 Purpose

The LMT naming conventions serve to identify the LMT objects that can be managed via remote services on these objects.

6.2 LMT Object Name Syntax

A class 2 LMT Slave is identified by an LMT-Address. An LMT Address consists of a *manufacturer-name*, a *product-name* and a *serial-number*. The manufacturer-name and product name are symbolic names. The serial-number is a BCD number. They adhere to the following syntax:

```

<LMT-Address>      ::= <manufacturer-name> <product-name> <serial-number>
<manufacturer-name> ::= { <alpha-num> }7
<product-name>     ::= { <alpha-num> }7
<serial-number>    ::= { <bcd-pair> }7

```

A <manufacturer-name> is assigned to module suppliers by CiA. A <product-name> and a <serial-number> are assigned by the module supplier. For LMT-Addresses the following condition must be met:

- there exists no other class 2 LMT Slave in the world with the same <LMT-Address>